

## N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM  
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT  
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED  
IN THE INTEREST OF MAKING AVAILABLE AS MUCH  
INFORMATION AS POSSIBLE

OCTOBER 1979

(NASA-CR-161351) DATA BASE MANAGEMENT  
SYSTEM CONFIGURATION SPECIFICATION (General  
Electric Co.) 111 p HC A06/MF A01 CSCL 09B

N80-15825

Unclas  
G3/60 46603

# CONFIGURATION SPECIFICATION

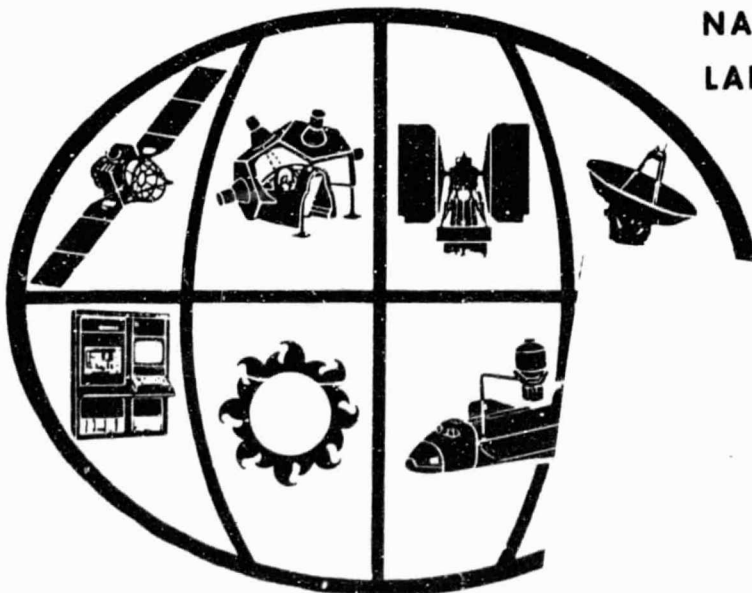


PREPARED UNDER

CONTRACT NO. NAS8-33374

NASA, MSFC, DATA SYSTEM

LABORATORY EF15



space division 

PREPARED BY  
THE GENERAL ELECTRIC CO  
HUNTSVILLE OPERATIONS  
OF THE SPACE DIVISION  
HUNTSVILLE, ALABAMA

GENERAL  ELECTRIC

REPORT NO 79HV013

OCTOBER 1979

DATA BASE MANAGEMENT SYSTEM CONFIGURATION SPECIFICATION

PREPARED UNDER CONTRACT NO. NAS8-33374

FOR

NASA/MSFC

DATA SYSTEMS LABORATORY/EF15

PREPARED BY:

James W. Neiers  
James W. Neiers  
Senior Engineer

CONCURRED BY:

D.W. Rowe  
D.W. Rowe, Manager  
Huntsville Operations

APPROVED BY:

Douglas T. Thomas  
Douglas T. Thomas  
Contract COR  
NASA/MSFC

GENERAL ELECTRIC COMPANY  
HUNTSVILLE OPERATIONS OF  
THE SPACE DIVISION  
HUNTSVILLE, ALABAMA

## TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
	LIST OF ILLUSTRATIONS . . . . .	v
	LIST OF TABLES . . . . .	vi
1	INTRODUCTION . . . . .	1
1.1	Background . . . . .	1
1.2	Program Elements . . . . .	1
1.3	Information Adaptive System . . . . .	2
1.4	Modular Data Transport System . . . . .	2
1.5	Data Base Management System . . . . .	2
1.6	Archival Mass Memory . . . . .	2
1.7	Objective of DBMS . . . . .	2
2	SYSTEM OVERVIEW . . . . .	4
2.1	System Data Flow . . . . .	4
2.2	System Functions . . . . .	8
2.3	Typical Data Flow From Staging Area Interface . . . . .	9
	2.3.1 Interface to MDTs . . . . .	9
	2.3.2 Information Frame Formats . . . . .	10
	2.3.3 Control of SAI Data Flow . . . . .	16
	2.3.4 Quick Look Mode . . . . .	17
2.4	Header Data . . . . .	17
	2.4.1 Header Transfer to Auxiliary Storage . . . . .	19
	2.4.2 Data Packet Identification . . . . .	19
	2.4.3 Packet Header Interface . . . . .	20
	2.4.4 Header Data Manager . . . . .	20
	2.4.5 Scenario for Header Data Processing . . . . .	21
2.5	Data Bus Transfers . . . . .	24
	2.5.1 Bus Control Concept . . . . .	25
	2.5.2 Control Time . . . . .	26
	2.5.3 Typical Class II Packet Transfer Scenario . . . . .	27
2.6	Staging Area Interface . . . . .	28
	2.6.1 Special Conditions . . . . .	28
	2.6.2 Level 4 Protocol . . . . .	28
	2.6.3 Typical Scenario of Receipt of Data . . . . .	29
2.7	Data Retrieval from AMM . . . . .	30
	2.7.1 Identification of Packets . . . . .	30
	2.7.2 Typical Sequence of Events in Data Retrieval . . . . .	30
	2.7.3 Error Conditions . . . . .	34
	2.7.4 Timing . . . . .	34
2.8	Typical Scenario of User Operation . . . . .	35
	2.8.1 User Terminal Function . . . . .	35
	2.8.2 Interrupt . . . . .	35
	2.8.3 Query . . . . .	36
	2.8.4 Data Base Processor Activity . . . . .	37
	2.8.5 Image Utilities . . . . .	37
	2.8.6 Additional Functions for the User . . . . .	39

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Title</u>	<u>Page</u>
2.9	DBMS Packets . . . . .	39
2.9.1	Data Blocks . . . . .	39
2.9.2	DBMS Packet Types . . . . .	40
2.10	Data Bus Function Codes . . . . .	40
3	HARDWARE SPECIFICATION . . . . .	43
3.1	Integrated Data Base Computer (VAX-I) . . . . .	43
3.2	Configuration Management Computer (VAX-II) . . . . .	46
3.3	Archival Mass Memory . . . . .	46
3.4	Auxiliary Storage . . . . .	46
3.5	Staging Area Interface Hardware . . . . .	49
3.6	User Terminals . . . . .	49
3.6.1	Alpha-Numeric Terminals . . . . .	49
3.6.2	Image Display . . . . .	51
3.7	Data Bus . . . . .	52
3.7.1	Data Bus Structure . . . . .	53
3.7.2	Data Bus Operation . . . . .	56
3.8	Master Controller and Data Bus Port . . . . .	57
3.8.1	Function . . . . .	57
3.8.2	Control Signals . . . . .	59
3.8.3	Special Hardware . . . . .	60
3.8.4	Microcomputer . . . . .	60
3.9	Typical User Data Bus Port . . . . .	60
3.9.1	Fiber Optic Interface . . . . .	60
3.9.2	Device Interface . . . . .	62
3.9.3	Master Controller Interface . . . . .	62
3.10	VAX Data Bus Port . . . . .	62
3.11	Auxiliary Storage Data Bus Port . . . . .	62
3.12	Archival Mass Memory Data Bus Port . . . . .	62
3.13	Staging Area Interface Data Bus Port . . . . .	65
3.13.1	Instrument Packet Port . . . . .	65
3.13.2	Instrument Packet Header Port . . . . .	65
3.14	Multiplexed Data Bus Port . . . . .	65
3.15	Multiplexer . . . . .	68
4	SOFTWARE SPECIFICATION . . . . .	70
4.1	Configuration Data Management Software . . . . .	70
4.2	Software Definition and Guidelines . . . . .	70
4.2.1	Operating System Software . . . . .	70
4.2.2	Support Processors . . . . .	70
4.2.3	Diagnostics . . . . .	70
4.2.4	Special System Software . . . . .	70
4.2.5	Applications Processes . . . . .	70
4.3	VAX I Operating System . . . . .	73
4.3.1	Standard Drivers . . . . .	73
4.3.2	Other Services . . . . .	73
4.3.3	Auxiliary Storage Driver . . . . .	73
4.3.4	Data Bus Port Driver . . . . .	74

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Title</u>	<u>Page</u>
4.4	VAX 11 Operating System . . . . .	75
4.4.1	Standard Drivers . . . . .	75
4.4.2	Other Services . . . . .	75
4.4.3	Master Controller and Data Bus Port Driver . . . . .	75
4.4.4	Image Display Driver . . . . .	76
4.5	Support Processors . . . . .	77
4.5.1	Standard VAX/VMS Support Processors . . . . .	77
4.5.2	Optional DEC Support Processor . . . . .	77
4.5.3	Microcomputer Support . . . . .	77
4.5.4	Master Controller and Data Bus Simulator . . . . .	78
4.5.5	Data Bus Port Simulator . . . . .	79
4.5.6	Other Device Simulators . . . . .	79
4.5.7	Display Processors . . . . .	79
4.6	Diagnostics . . . . .	79
4.6.1	Computer and Standard Peripheral Devices . . . . .	80
4.6.2	Data Bus Port . . . . .	80
4.6.3	Data Bus . . . . .	80
4.6.4	Archival Mass Memory . . . . .	80
4.6.5	Auxiliary Storage . . . . .	80
4.6.6	Display System . . . . .	81
4.6.7	Staging Area Interface . . . . .	81
4.7	Special System Software . . . . .	81
4.7.1	Integrated Data Base Management System . . . . .	82
4.7.2	Configuration Management System . . . . .	83
4.7.3	System Data Tables . . . . .	84
4.7.4	Verification Software . . . . .	88
4.7.5	System Data Table . . . . .	89
4.7.6	Verification Software . . . . .	91
4.8	Software Overview . . . . .	94
4.8.1	Typical Data Flow from DBMS to Retrieve AMM Data . . . . .	96
4.8.2	Typical User Terminal Interfaction with DBMS . . . . .	99
4.9	Software System Interface . . . . .	101
5	OTHER FACTORS . . . . .	102
5.1	Reliability . . . . .	102
5.2	Maintainability . . . . .	102
5.3	Workmanship . . . . .	102
5.4	Quality Assurance Provisions . . . . .	102
5.5	Documentation . . . . .	103
5.5.1	Specially Developed Equipment . . . . .	103
5.5.2	Commercially Purchased Equipment . . . . .	103
5.5.3	Software . . . . .	103
5.6	Testing . . . . .	103
	LIST OF ACRONYMS . . . . .	104

## LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
2-1	Structure of the Data Base Management System . . . . .	5
2-2	DBMS Hardware Configuration . . . . .	6
2-3	DBMS Data Flow . . . . .	7
2-4	Data Formats Received at X.25 Interface . . . . .	11
2-5	Data Formats Transferred to Archival Mass Memory . . . . .	18
2-6	Process Flow for Header Data . . . . .	22
2-7	Data Bus Time Slots . . . . .	25
2-8	Data Bus Control Time Slot . . . . .	25
2-9	DBMS Packet Format . . . . .	41
3-1	DBMS Hardware Configuration . . . . .	44
3-2	VAX-I Functional Block Diagram . . . . .	45
3-3	VAX-II Hardware Configuration . . . . .	47
3-4	Auxiliary Storage Hardware Configuration . . . . .	48
3-5	Staging Area Interface Hardware Functional Block Diagram .	50
3-6	Data Bus Timing Cycle . . . . .	54
3-7	Class 1 and Class 2 Data . . . . .	55
3-8	Time Division Multiplexed Data Bus . . . . .	56
3-9	VAX Data Bus and Master Controller . . . . .	58
3-10	User Terminal Data Bus Port . . . . .	61
3-11	VAX Data Bus Port . . . . .	63
3-12	Auxiliary Storage Data Bus Port . . . . .	64
3-13	Archival Mass Memory Data Bus Port . . . . .	66
3-14	Instrument Packet Data Bus Port . . . . .	67
3-15	Multiplexed Data Bus Port . . . . .	69
4-1	Data Bus Processing Overview . . . . .	85
4-2	DBMS Software. . . . .	95
4-3	Typical Data Flow from IDBMS to Retrieve AMM Data . . . .	97
4-4	User Terminal Interfaction with IDBMS . . . . .	100

## LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
2-1	Tabulation of Source ID Codewords in Hexadecimal Notation which Maximizes the Minimum Distance . . . . .	15
2-2	Packet Identification . . . . .	27
2-3	DBMS Packet Types . . . . .	42
2-4	Bus Function Codes . . . . .	42
4-1	Bus Function Codes . . . . .	91
4-2	Master Controller Interrupt Process . . . . .	99
4-3	Sequence for Master Controller Processing Command from UT1 to Establish Communication . . . . .	101



## SECTION 1

### INTRODUCTION

This specification describes the functional requirements and the configuration of the Data Base Management System (DBMS) that will be used in conjunction with other systems comprising the NASA End-to-End Data System (NEEDS Phase 2) Program to demonstrate techniques and technology which will enable more efficient and timely transfer of useful data from the sensor to the user, extraction of information by the user, and exchange of information among the users.

#### 1.1 BACKGROUND

The space program of the 1980's and beyond is addressing the nation's needs through an extensive program involving Spacelab, earth orbital satellites, and planetary probes. The quantity and complexity of data envisioned from these sources is staggering, with the cost of data acquisition and distribution presenting a major problem. NASA has set a goal of 10X cost reduction and 1000X increase in information return. This program supports those goals by providing a data system architecture which lends itself to multi-mission applications, increased onboard autonomy, decreased software costs, and simplified user interaction.

#### 1.2 PROGRAM ELEMENTS

The NEEDS Phase 2 Program comprises several elements including: Information Adaptive System (IAS), Modular Data Transport System (MDTS), Data Base Management System (DBMS), Massively Parallel Processor (MPP) and Archival Mass Memory (AMM).

The DBMS includes the Integrated Data Base Management System (IDBMS), a software system under development at GSFC that will operate in the DBMS, and the AMM being developed at MSFC. The DBMS has a functional interface with the IAS and a physical and functional interface with the MDTS.

### 1.3 INFORMATION ADAPTIVE SYSTEM

The IAS will develop and demonstrate onboard spacecraft capability for adaptive control and processing of sensor data. Onboard data calibration and preprocessing will reduce the cost of ground data handling.

### 1.4 MODULAR DATA TRANSPORT SYSTEM

The MDTs will provide a packetized delivery of the space data to the DBMS. The MDTs will insure the integrity of the delivered data as well as perform the necessary reformatting to accommodate the various modes of delivery, such as space to ground, or ground to ground.

### 1.5 DATA BASE MANAGEMENT SYSTEM

The DBMS will provide storage and retrieval of space data using techniques that provide a friendly interface with a broadly based user community. The DBMS will receive data from the MDTs-Staging Area and archive it. It will provide procedures, formatting, and retrieval methods to enhance multiple user access to instrument data with unique data and format characteristics. It will also provide archiving and retrieval of information extracted from space data and collateral space data.

### 1.6 ARCHIVAL MASS MEMORY

The AMM is a high density archival storage device capable of managing the long term archival and retrieval of  $10^{13}$  bits. It will be incorporated into the DBMS as Government Furnished Equipment (GFE). It will store and retrieve packets of data ranging in size from 256 bits to 8 megabits (8388577 bits).

### 1.7 OBJECTIVE OF DBMS

The primary objective of this system, DBMS, is to demonstrate the technology necessary to archive large volumes of data at high data rates in near real time, to catalog and create a directory of the data based on available information about the data, and to make the directory and data available to the user in a timely manner. The vehicle by which information may be extracted from the data will be available to the user. The degree of information extraction, however, would be determined by a data base administrator. The application of

the technology includes: global access of the user to relevant sensor data, data bases, information bases, and other system users. Toward this objective, the system specified herein shall emphasize techniques, data rates, and technology that will offer growth capacity.

## SECTION 2

### SYSTEM OVERVIEW

This section provides a general description of the Data Base Management System requirements in a summary form. The intention of these requirements is to provide an overview of the composite system and the DBMS demonstration environment. All requirements in this section shall be considered specifications even though they may be repeated in the detail specifications.

The DBMS comprises the hardware and software elements of Figure 2-1. The major hardware elements are: (1) VAX I, (2) VAX II, (3) Auxiliary Storage (AS), (4) Archival Mass Memory (AMM), (5) Staging Area Interface (SAI), (6) User Terminals (UT), and (7) Data Bus. The system computers, VAX I and VAX II are Digital Equipment Company (DEC) VAX 11/780 minicomputers. VAX I is an existing Government-furnished system that will primarily function as a host for the Integrated Data Base Management Software System. VAX II will be an additional comparable system with the principal function of executing the Configuration Management System (CMS). The AMM is Government Furnished Equipment (GFE) procured on a separate contract. The user terminal display and auxiliary storage will be commercially available equipment. The Data Bus and the Staging Area Interface will be developed specifically for the DBMS. The physical relationship of each element is shown on the "DBMS Overall System Diagram," Figure 2-2.

The software comprises that available or existing with the computer system, special drivers, supporting software, and two major systems: the IDBMS and the CMS. The IDBMS is being developed by and will be installed on VAX I by GSFC. The CMS shall be specially developed for DBMS.

#### 2.1 SYSTEM DATA FLOW

Data Flow within the DBMS is illustrated by the diagram of Figure 2-3. Data will be received at the staging area interface in packets ranging in size from 256 bits to 8 megabits. The packets, called instrument packets (IP), will contain header information that will uniquely identify the data for subsequent management by the DBMS. The IP's, including the header information, will be transferred to the AMM for archiving. The header data in excess of 2048 bits,

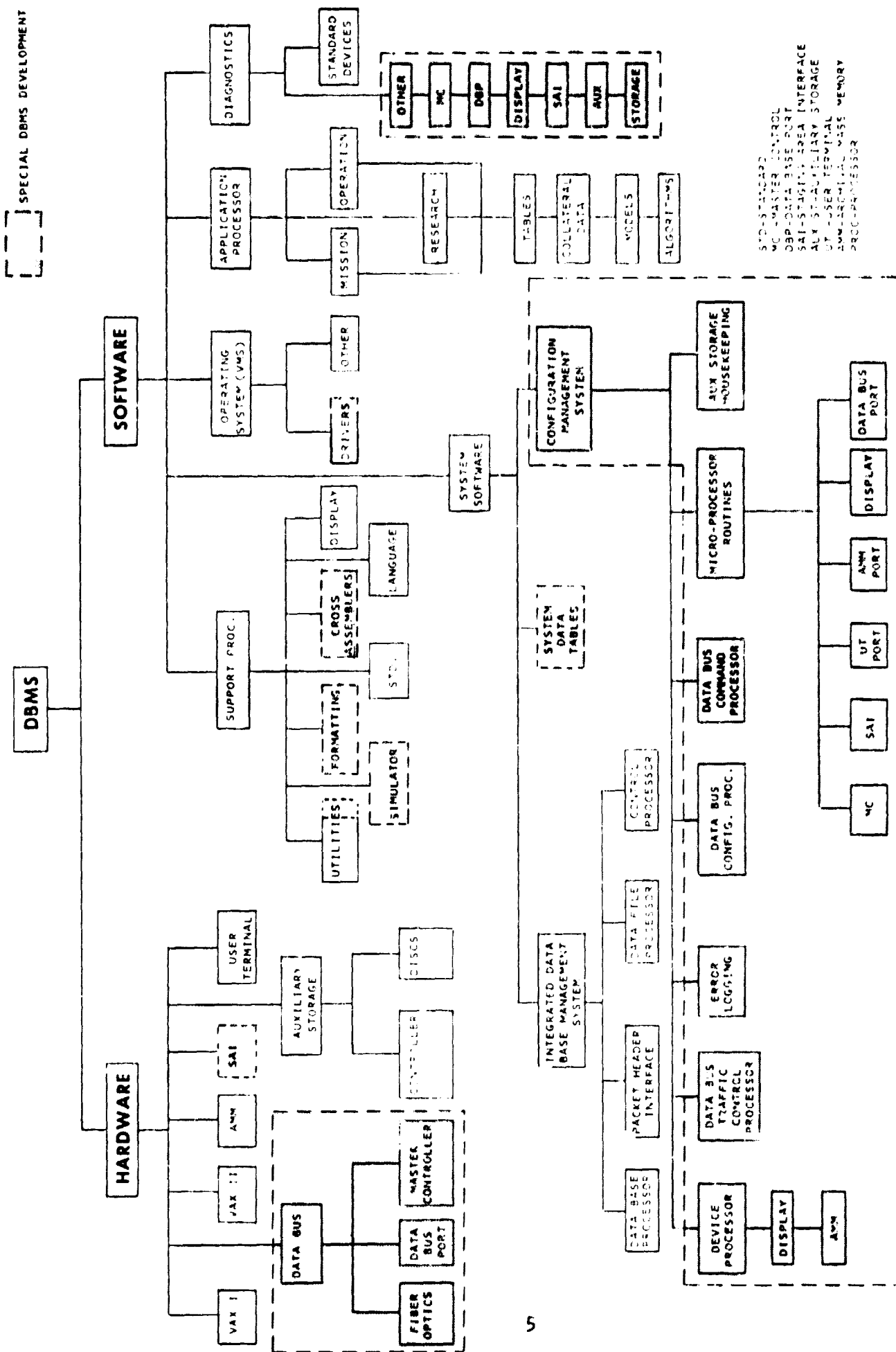
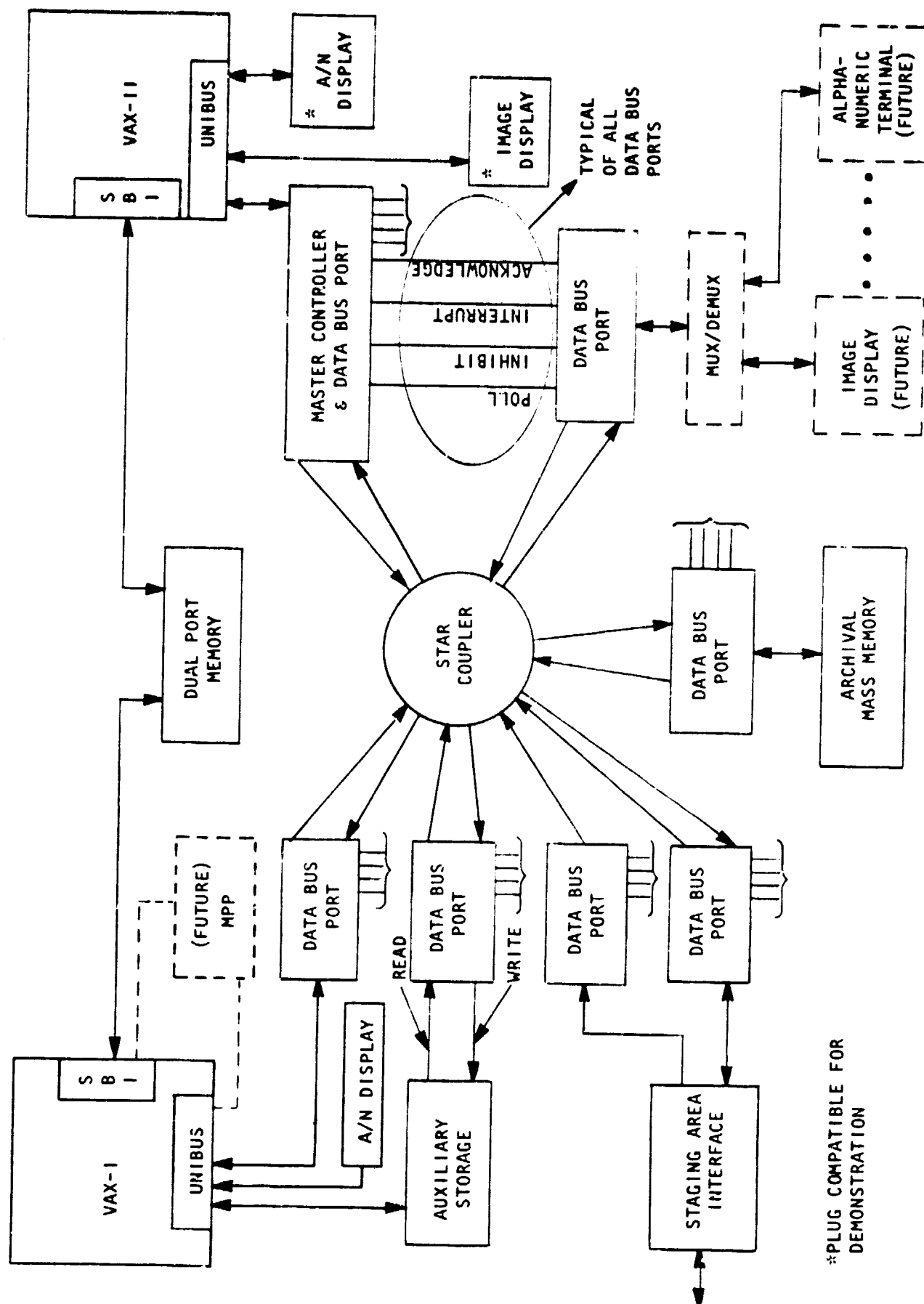


Figure 2-1. Structure of the Data Base Management System



**Figure 2-2. DBMS Hardware Configuration**

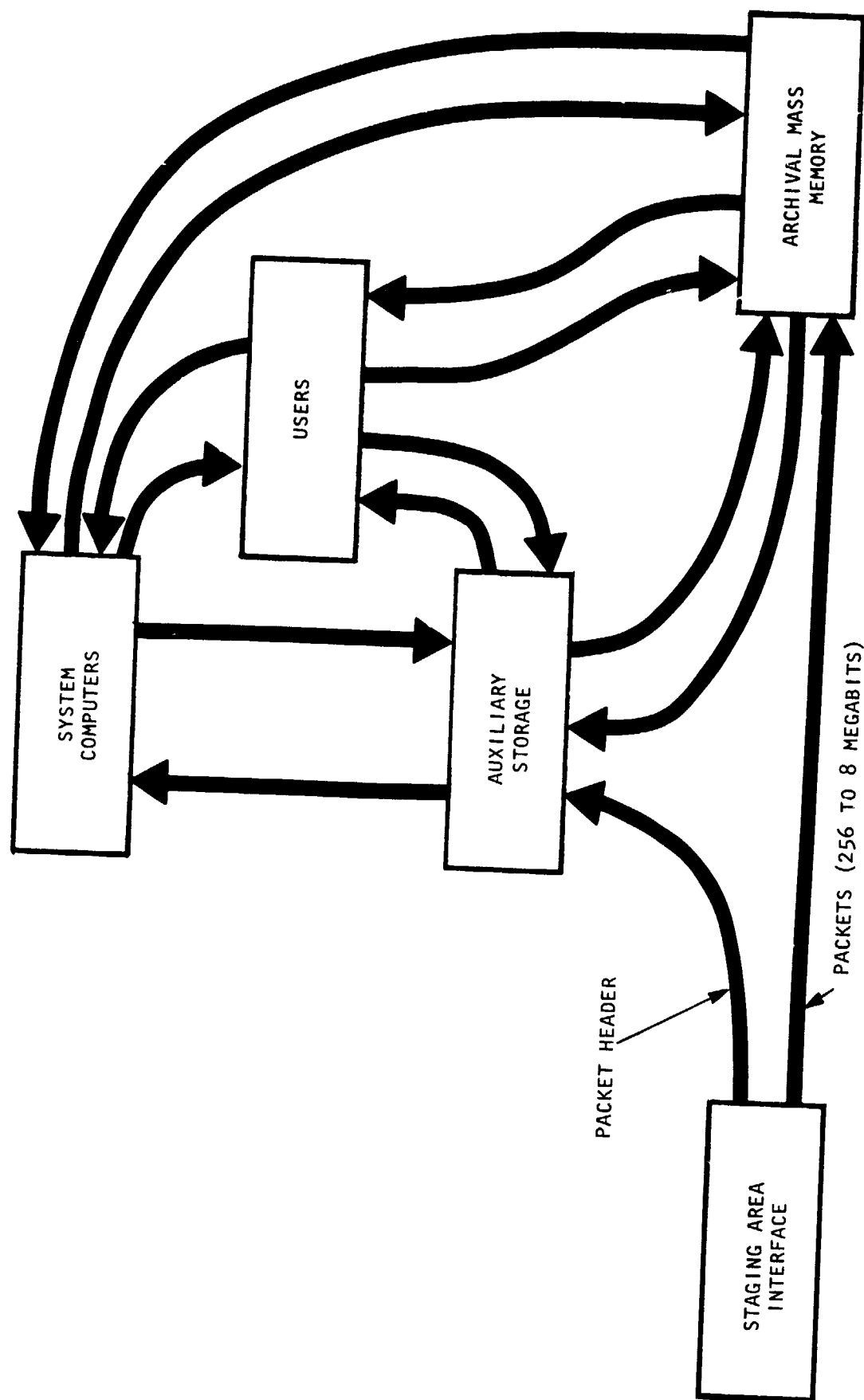


Figure 2-3. DBMS Data Flow

the standard system block, will not be routed directly to the auxiliary storage but may be retrieved from the AMM as required during the establishment of relational tables.

Data transfers shall occur between the SAI and the AMM without interference from any other data transfers in the system. The CMS software operating in VAX II shall monitor the transfers of instrument packets to the AMM and shall not initiate any other data transfers to the AMM while the staging area interface is active. Data flow between the SAI and the AMM shall be autonomously initiated by the SAI. All other data transfers within the system shall require the initiation of the packet transfer by the CMS software operating in VAX II. Transfers of blocks of data shall be under the control of the master controller.

Data packets of origin other than the staging area shall also be accommodated within the DBMS. Such data, called DBMS packets, may originate from, but not be limited to, relational tables generated within the Integrated Data Base Management Software (IDBMS), overflow from the auxiliary storage, or the results of user processing. The DBMS packets shall have sufficient header data as defined in paragraphs 2.4.2 and 2.9.2 to insure subsequent management and retrieval.

The system shall provide for direct data transfers to and from the AMM and the user terminals, the auxiliary storage, and the system computers without requiring an intermediate transfer to the memories of VAX I, VAX II, or the auxiliary storage. All of these transfers, called bus transfers, shall take place without interference with the transfer of data between the SAI and the AMM except that data shall not be simultaneously transferred to the AMM from the bus and the SAI. Adequate conflict resolution shall be provided with a priority given to the SAI source. The system shall provide for the direct transfer of data between VAX I and auxiliary storage via the Unibus.

## 2.2 SYSTEM FUNCTIONS

The DBMS shall provide the following primary functions:

1. Receive IP's at the SAI according to CCITT.X.25 protocol, hereafter called X25.



2. Verify the integrity of the received data frame according to X25 protocol.
3. Interpret and format the IP header for data management.
4. Archive the IP's.
5. Maintain adequate cognizance of the archived packets, both IP's and DBMS packets, for subsequent retrieval.
6. Provide for the maintenance of user friendly retrieval techniques.
7. Provide assistance and user interaction for retrieval of information.
8. Provide utility operations to reformat, edit, and display the retrieved data.
9. Provide a host for the execution of a limited amount of user furnished information extractions software.
10. Provide for a limited amount of image display processing.

### 2.3 TYPICAL DATA FLOW FROM STAGING AREA INTERFACE

In a typical operation of the DBMS performing the archival of space data function, the SAI will be monitoring the communications link. According to established X.25 protocol, the conditions of an available non-busy channel will be discerned. The receipt of appropriate supervisory packets will signal the intent of the staging area to initiate an IP transfer and it will elicit the appropriate SAI responses. The packetization protocol of X25 permits considerable flexibility in the transfer of space data to the DBMS.

#### 2.3.1 INTERFACE TO MDTs

The interface between the MDTs and the DBMS shall be in accordance with CCITT X.25 "Interface Between Data Terminal Equipment (DBMS) and Data Circuit - Terminating Equipment for Terminals Operating in the Packet Mode on Public Data Networks." This provides a level 3, i.e., at the IP level, interface with the Staging Area of the NEEDS/MDTs and a level 2, i.e., X25 frame level, with the communication equipment at the adjacent node to the DBMS node. Each IP shall be transferred as a single X25 level 3 packet that shall be a minimum of 256 bits and a maximum of 8 megabits. Each X25 level 2 frame shall be a minimum of 48 bits and a maximum of 2096 bits.

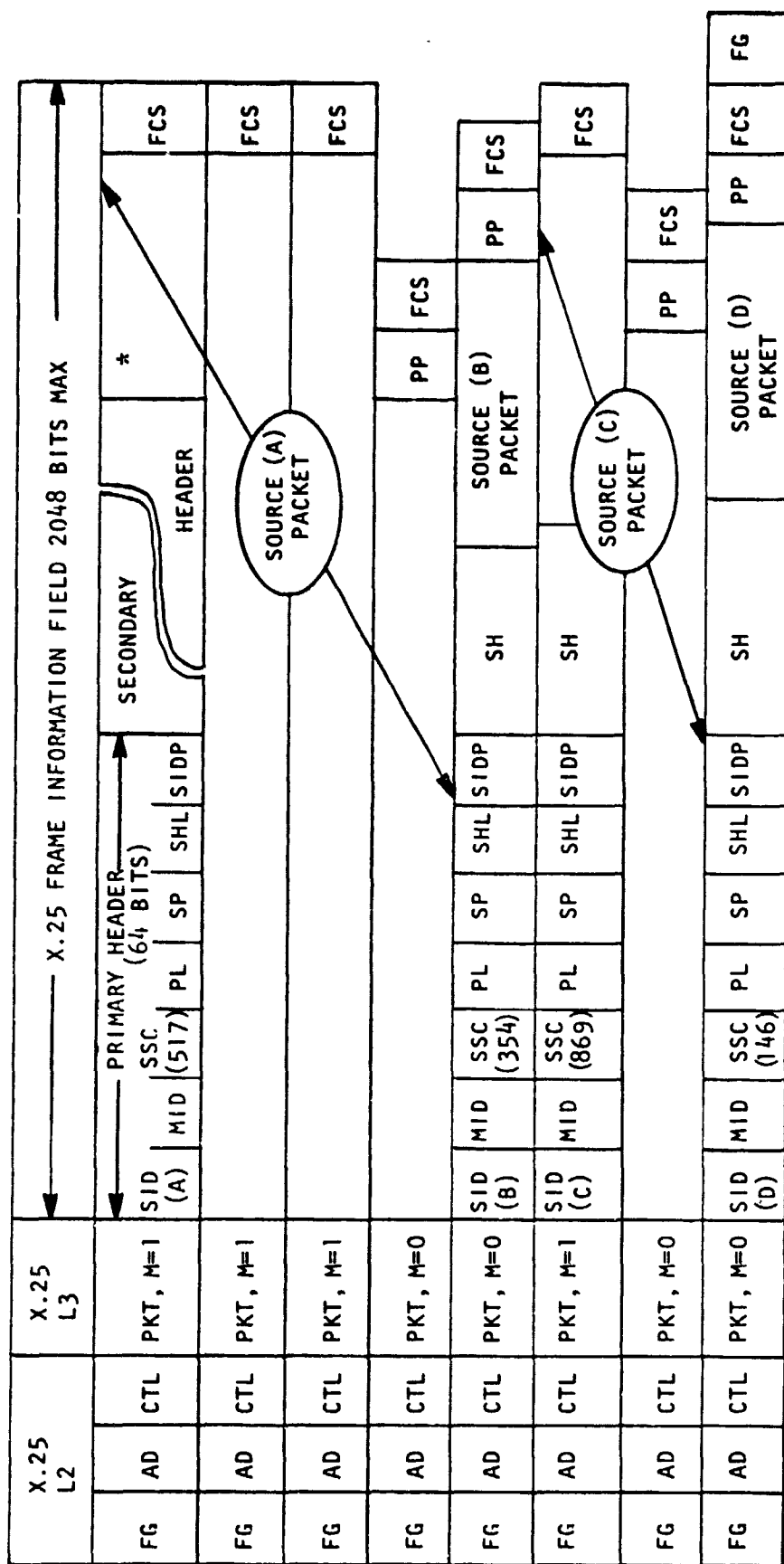
According to the X25 protocol, frames of data will be either supervisory or information and the control field will be formatted accordingly. The supervisory frames provide for the necessary communication protocol for establishing and verifying the channels. The control field for the information frames provides for the necessary routing, identification and validation of the communications. The specification of the supervisory frames and X25 level 4 protocol is specified in paragraphs 2.6.2 and 2.6.3. Information frames are specified in the following paragraph.

### 2.3.2 INFORMATION FRAME FORMATS

A typical set of X.25 information frames as received at the SAI is shown in Figure 2-4. This figure illustrates the DBMS interface with the other NEEDS elements MDTs and IAS. The first column, entitled "X.25 L2" is the level 2 interface which contains: 1) the flag which identifies the start and end of each frame, 2) the address which is used by the communications network to route the frame, and 3) the control which identifies the type of frame, either supervisory or information, and for information frames, an identification of the sequence and acknowledgements. The second column, entitled "X.25 L3", is the level 3 interface with the MDTs Staging Area. There is a one-to-one correspondence between this interface and the source packets or IP's. A single bit known as the multiple bit in this field signifies that additional frames are required to complete the IP by being set "M = 1." When no additional frame is required, "M = 0." The IP, which is a level 6 interface with the IAS, is embedded in the third column entitled "X.25 Frame Information Field 2048 Bits Max." The last 16 bits of this field are inserted as part of X.25 level 2 protocol to provide a frame checking sequence (FCS) or cyclical redundancy check (CRC) on the data frame.

#### 2.3.2.1 Instrument Packet

The IP may be an information packet originating onboard a spacecraft or from some ground processing location. It may contain instrument data or it may be a utility packet containing data about an instrument or process. It will always contain a header and a packet parity (PP) field. The header will always commence with the first frame (L2) and the PP will always be in the final frame.



LEGEND:	FG	FLAG 01111110 - IDENTIFIES X.25 FRAME	PL	PACKET LENGTH (PL)
	AD	ADDRESS (8 BITS)	SP	SPARE (8 BITS)
	CTL	CONTROL (8 BITS) - IDENTIFIES TYPE	PP	PACKET PARITY
		FRAME & COUNT MODULO 8	SH	SECONDARY HEADER
	PKT	PACKET (24 BITS) - IDENTIFIES X.25	SHL	SECONDARY HEADER LENGTH (8 BITS)
		PACKET	SIDP	SOURCE ID PARITY (8 BITS)
	SID	SOURCE ID (8 BITS)	FCS	FRAME SEQUENCE CHECK (16 BITS)
	MID	MISSION ID (8 BITS)		
	SSC	SOURCE SEQUENCE COUNT (16 BITS)		

\* THE SECONDARY HEADER LENGTH MAY EXCEED THE X.25 FRAME LENGTH.

**Figure 2-4. Data Formats Received at X.25 Interfac3**

The header will always contain 64 bits\* designated the "primary header" and it may contain additional information designated the "secondary header." The secondary header may range in length from zero bits to the maximum packet length.

For packets requiring more than one frame, each frame except the last shall be 2048 bits long including the FCS. The last frame of multiple frame packets may have an information field as short as 17 bits, i.e., 1 bit of the continuation of packet parity data and 16 bits of FCS. A single frame packet shall have an information field at least 256 bits long to coincide with the minimum length IP.

Each IP, i.e., level 3 packet, shall be transferred in its entirety by a contiguous sequence of level 2 frames.

#### 2.3.2.2 Primary Header

The primary header consists of 64 bits\* which provide the source and sequence identification and the length of the overall packet and its secondary header. The fields which make up the primary header are as follows:

<u>Field</u>	<u>Abbreviation</u>	<u>Length (bits)</u>
Source ID	SID	8
Mission ID	MID	8
Source Sequence Count	SSC	16 (32)*
Packet Length	PL	8
Spare	SP	8
Secondary Header Length	SHL	8
Source ID Parity	SIDP	<u>8</u>
Total		64 bits*

\* Optionally may increase by 16 bits.

The source ID is an 8-bit field uniquely identifying the instrument assembly or spacecraft subsystem within a mission which is the origin of the current source packet. The mission ID is an 8-bit field uniquely identifying the current mission. In the case of reusable vehicles (ST, Spacelab, etc.), a new Mission ID number will be assigned for each launch or refurbishment.

The Source Sequence Count is a 16-bit field representing a sequentially incrementing binary count (modulo  $2^{16}$ ) of the number of IP's generated by the specified source. This number is used by the DBMS in the establishment of relational tables for data sets consisting of multiple IP's. There is a possibility the length of this number will be increased for future operational conditions to insure a unique packet identification.

The normal format of the 8-bit packet length field defines the packet length in a floating point representation. The first four bits of the field represent the exponent (E) and the last four bits represent the mantissa (M). The specified packet length (L) in bits is given by Equation 2.1.

$$L = \left( \frac{1}{2} + \frac{M}{32} \right) \times 2^{E+8} \quad (2.1)$$

$$M = 0, 1, \dots, 15$$

$$E = 0, 1, \dots, 14$$

Although a wide range of packet lengths are defined by this normal packet length format, it is envisioned that the earlier packet telemetry missions will use packet lengths on the order of 4,000 bits. To provide optimal packing within the present NASCOM 4,800-bit block format, JPL is anticipating using packet lengths of either 4,480 or 4,560 bits. Neither of these lengths, nor those exceeding 2,031,616 bits, can be specified by the length algorithm given by Equation 2.1. To accommodate up to 16 additional special packet lengths which may be specified in the future, the 16 packet length codes which begin with four "L" bits (E-15) are not defined by Equation 2.1, but can be allocated by the Director of the National Space Science Data Center (Code 601) as the need arises. Whenever practical, the use of special packet lengths should be avoided since externally supplied information must be provided for their interpretation.

The 8-bit spare field is presently unassigned. The 8-bit secondary header length code specifies the number of 8-bit bytes in the secondary header. The secondary header immediately follows the primary header field. The algorithm to be employed in computing the secondary header length is TBD. It will probably be some floating point number to accommodate secondary header length of greater than 2048 bits.

The 8-bit Source ID Parity field represents a code which is a redundant specification of the Source ID field. The Source ID field and the Source ID Parity field form a systematic binary (16, 8) cyclic block code with generating polynomial  $G(X) = X^8 + X^4 + X^3 + 1$ . Each source instrument will be assigned a fixed unique 16-bit code word so that it is not necessary (or desirable) for the source instrument to independently compute the 8 parity bits inserted in this field. The 256 valid code words of this code are listed in Table 2-1. The minimum Hamming distance between valid code words is 5; hence, this code is inherently capable of correcting one or two random bit errors anywhere within the 16-bit encoded block. The 16-bit Source ID code words should be assigned to the different instrument assemblies in the order specified in Table 2-1. However, in standard applications, error detection only will be implemented.

#### 2.3.2.3 Secondary Header

The purpose of the secondary header is to provide a means for encoding within a source packet any ancillary data (time, position, attitude, etc.) which may be necessary for the interpretation of the source data. A "Table of Contents" field (in a format to be determined) will be included as the first field of the secondary header and will define the types and formats of the ancillary data contained within the secondary header.

#### 2.3.2.4 Utility Packets

Utility Packets provide an alternative means of correlating ancillary data with sensor data. The ancillary data will principally contain time and location information. The nature of the location information will depend upon the degree of onboard processing that is implemented. Currently, locating information includes spacecraft ephemeris and attitude and any sensor pointing values that are applicable. Other sensor engineering data such as emplaced

**PACKET DATA GUIDELINE**  
**1979-05-04**

**Table 2-1. Tabulation of Source ID Codewords in Hexadecimal Notation Which Maximizes the Minimum Distance**

START BY SELECTING THE FOLLOWING 2 CODEWORDS WITH MINIMUM DISTANCE 12									
0000	3778								
IF NECESSARY, ADD FOLLOWING 2 CODEWORDS TO GET A TOTAL OF 4 CODEWORDS WITH MINIMUM DISTANCE 16									
00A7	588C								
IF NECESSARY, ADD FOLLOWING 4 CODEWORDS TO GET A TOTAL OF 8 CODEWORDS WITH MINIMUM DISTANCE 8									
50EC	6707	B177	B6AC						
IF NECESSARY, ADD FOLLOWING 8 CODEWORDS TO GET A TOTAL OF 16 CODEWORDS WITH MINIMUM DISTANCE 7									
1861	2C9A	4681	71NA	9A16	ADED	CAEA	FD11		
IF NECESSARY, ADD FOLLOWING 8 CODEWORDS TO GET A TOTAL OF 24 CODEWORDS WITH MINIMUM DISTANCE 6									
143F	23C4	580D	6FF6	8E29	B902	C218	F5ED		
IF NECESSARY, ADD FOLLOWING 32 CODEWORDS TO GET A TOTAL OF 256 CODEWORDS WITH MINIMUM DISTANCE 5									
0139	0272	034B	0457	05DD	0696	079F	08F1	09C8	0B8A
0D2C	0E67	0F39	1053	11E5	12AF	139F	1571	1646	172A
2157	230C	2457	2635	2865	2AFA	2C01	2E04	2F60	3110
3359	35C2	37C5	399C	3C78	3F66	42F7	4684	4A93	4E8C
51C5	5418	5724	5A77	5D92	6080	656A	6A53	6F39	7326
7693	79AB	7C66	7F41	82F2	87D3	8C34	8F38	9275	976C
992C	9A48	9B71	9C98	9E3A	AACD	AD34	AE59	B032	B34A
BA79	BB00	BC1C	BD35	BE7A	C07A	C38B	C694	C9A7	CC2A
CE54	D071	D382	D693	DA34	DD4A	E05B	E36C	E67D	F18E
F499	F7A0	F8B1	F9C2	FA70	FB76	FC49	FE68	F792	F86E

**NOTE: 1.) The first two and the last two hex characters will be assigned to the Source ID and Source ID Parity fields respectively.**

2.2.) If desired, any fixed 16-bit sequence can be modulo 2 added (Exclusive OR) with all the codewords without disturbing the relative Hamming distance property.

filters, detector temperatures, power supply values, bias currents, applied smart sensor algorithms, encoding, filtering, companding techniques, etc., can be expected in the ancillary data. The specification of the utility packet format will be the subject of a future standard.

#### 2.3.2.5 Packet Parity

The L-bit source packet is encoded into a systematic (L, L-16) binary cyclic block code using the ADCCP generating polynomial  $G(X) = X^{16} + X^{15} + X^2 + 1$ . The 16 parity bits are included as the final 16 bits of the packet. This code will be implemented for error detection only.

#### 2.3.3 CONTROL OF STAGING AREA INTERFACE DATA FLOW

Primary control of the data flow from the SAI will be autonomously controlled by the SAI. Initially the system shall be capable of accepting data at the SAI at 50 megabits per second. As a design goal, higher rates of 100 megabits shall be accommodated by the basic system without requiring redesign.

Data arriving at the SAI in 2K bit frames shall be buffered sufficiently to permit validation at the frame level. Because of the possibility of packets up to 8 million bits in length, it will not be necessary to maintain strict separation and transparency between the DBMS and MDTs. Validated data frames may be transferred to the AMM for archiving prior to the receipt of the complete packet. This could result in the archiving of bad data. The preamble incorporated in each instrument packet at the staging area as described in paragraph 2.3.3.1 shall be modified for each successive duplicated instrument packet to insure a unique packet identification code. The presence of non-zeros in bits 2 to 7 of the DBMS header ID field will serve as a mark for the IDBMS software in establishing the retrieval tables. The SAI will increment the count for each successive duplicated packet. Retrieval will always be based on the latest and highest count identifier. The AMM will effectively ignore the bad data because there will be no retrieval commands generated for it by the IDBMS. Additional description of the identification of redundant data packets is provided in paragraph 2.4.2. Protocol at the link level between the SAI and the AMM shall be maintained. The data bus shall be designed such that other data transfers on the bus will not interfere with the receipt and transfer of data from the SAI at 50 megabits per second.



#### 2.3.3.1 IP's to Archival Mass Memory

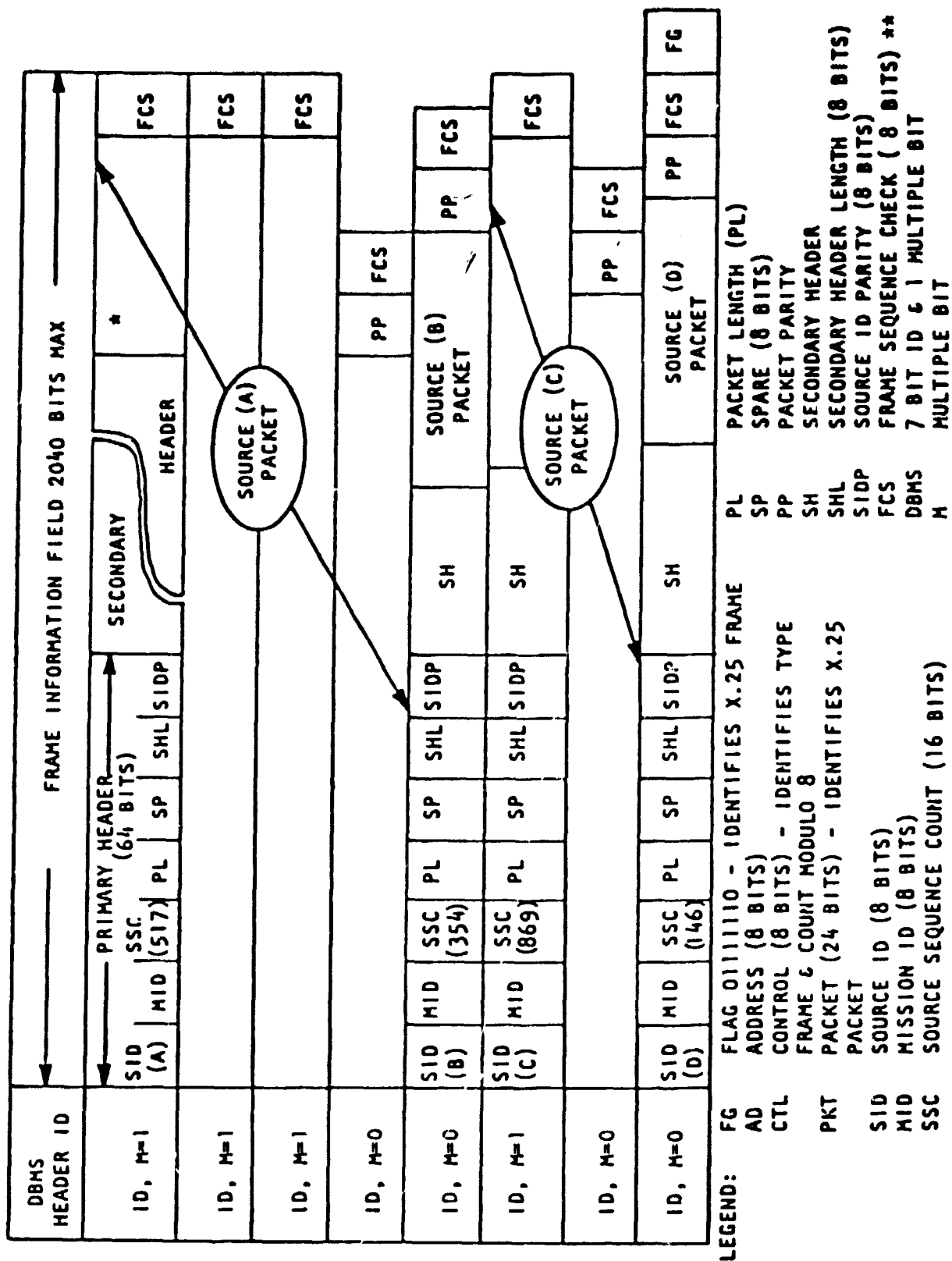
The entire IP shall be transferred to the AMM in 2K block increments. Prior to transfer, the X.25 protocol shall be removed, 8 bits of preamble shall be generated, and a new 8-bit CRC shall be generated for each block. The IP data shall not be altered. Typical data formats transferred to the AMM are shown in Figure 2-5. These are typical of all the IP block transfers within the DBMS. The first bit of the preamble shall identify the packet (block) as either staging area origin (1) or internal DBMS origin (0). The last bit shall be used as a multiple block for instrument packet indication such as in the level 3 X.25 protocol. The other six bits shall be reserved for future identification of the transfer source.

#### 2.3.4 QUICK LOOK MODE

The DBMS shall provide for a quick look mode of data transfer. In this mode, the entire designated IP shall be made available at the auxiliary storage within 30 seconds of its receipt. The direct transfer of the IP to the auxiliary storage in addition to AMM shall be permitted. An alternate method of implementation is the immediate transfer of the IP from AMM to Auxiliary storage prior to archiving. A degradation of the rate of acceptance at the SAI will be permitted when the DBMS is operated in this mode.

#### 2.4 HEADER DATA

The header data from each IP shall be placed in a pre-established file in auxiliary storage for use by the IDBMS. When initiated, the Packet Header Interface (PHI) will read the header data in this temporary file, interpret it, and use it to establish the relational tables and files required for the IDBMS to manage the data base. The CMS will purge the temporary file in the auxiliary storage upon completion of the PHI activity by IDBMS. References are made in the following paragraphs to the Data Base Processor (DBP), and the Data File Processor (DFP) that are part of the IDBMS. These and other software processors are discussed in paragraphs 4.7.1 through 4.7.1.3.



\* THE SECONDARY HEADER LENGTH MAY EXCEED THE FRAME LENGTH.

\*\* MODIFIED AFTER RECEIPT AT I/F.

Figure 2-5. Data Formats Transferred to Archival Mass Memory

#### 2.4.1 HEADER TRANSFER TO AUXILIARY STORAGE

The transfer of header data to auxiliary storage shall be initiated as soon as the frame is determined to be valid and will occur under control of the data bus Master Controller. If the frame containing header data must subsequently be repeated due to other failures, the transfer to auxiliary storage shall be repeated. Software handling the headers shall recognize and resolve duplication.

Only the first frame of header data needs to be transferred directly to the auxiliary storage. For longer headers, which are permissible but not common, subsequent retrieval of the header shall be accomplished from the AMM. For bursts of data for which the header data cannot be accommodated in real time, such as a sequence of short packets, the X25 protocol as implemented by the staging area interface will effectively delay the receipt of the transmission to a workable level, thus preventing any data loss.

As soon as a block (frame) of header data is ready for transfer to auxiliary storage, an interrupt signal shall be placed on the VAX Unibus. The VAX shall notify the auxiliary storage of a high priority transfer. As soon as the receiving buffer is ready and the bus is free as determined by the bus controller, the SAI shall be notified by the Unibus to commence transfer. The transfer shall continue until the auxiliary storage acknowledges receipt of the valid block of data. The Unibus shall then be notified.

#### 2.4.2 DATA PACKET IDENTIFICATION

Each data packet archived in the DBMS shall be uniquely identified by the first 40\* bits of data in the primary header. The first 8 bits shall be used by the CMS to identify the packet as either an IP from the staging area or of internal DBMS origin. The remaining bits shall identify the origin of the data to the source, mission, and sequence level or equivalent. The following allocation of identification bits is currently valid for IP's originating from the SAI.

---

\* Optionally 56 bits within the first 88 bits.

<u>Bits</u>	<u>Allocation</u>	<u>Number of Bits</u>
1	Identifies Source as Staging Area or DBMS	1
2 - 7	DBMS Packet Types (see Table 2-3)	6
8	Multiple Bit	1
9 - 16	Source ID or equivalent	8
17 - 24	Mission ID or equivalent	8
25 - 40	Source Sequence Count or equivalent	<u>16*</u> (32)
Total		40

For packets originating at the staging area, as indicated by a 1 in the bit 1 position, the next six bits in the DBMS field shall be used to discriminate between packets with the same source ID, mission ID, and source sequence count that were retransmitted because of error conditions. The six bits in the DBMS field shall be incremented for each successive retransmission. The low order count shall be a one in bit seven which shall be the least significant bit. Some minor deviations of the above allocation may be expected for DBMS packets. The total number of bits used to uniquely identify a data packet shall not exceed 56.

#### 2.4.3 PACKET HEADER INTERFACE (PHI)

The PHI is a software processor that is a part of the IDBMS. It shall provide for an automatic and a manual mode of operation. Its primary function is to access software routines of the IDBMS to maintain the necessary files and tables. These include both the relational tables used by the Data Base Processor and the non-relational tables used by the Data File Processor.

#### 2.4.4 HEADER DATA MANAGER (HDM)

The HDM is a software processor that is a part of the CMS. It shall provide two functions. It initiates the PHI and it purges the temporary file space in auxiliary storage after the PHI has completed its tasks and no longer needs the header data. It also manages the storage of header data in the auxiliary storage subsystem. The HDM computes and maintains the starting address of each IF header data in auxiliary storage.

---

\* Optional

## 2.4.5 SCENARIO FOR HEADER DATA PROCESSING

Each time an IP is archived, its header will be placed in a temporary file in auxiliary storage according to paragraph 2.4.1. Options shall be permitted as to the frequency of processing the header data. It is desirable to maintain currency in the Relational Tables of all packets archived. However, for some data sets such as image data, several IP's are required. For data sets requiring multiple IP's, it is desirable to wait until all of the IP's are archived before processing the header data. A candidate algorithm for determining the time to process header data is provided in paragraph 2.4.5.1. The relationship of the flow for processing header data is illustrated in Figure 2-6.

### 2.4.5.1 Initiation of the Packet Header Interface

The Header Data Manager shall initiate the PHI according to an algorithm that shall 1) prevent overflow of header data beyond the capacity of the allocated file space, 2) prevent the elapse of excessive time before processing, and 3) balance the requirements of 1 and 2 against the loading on the VAX for efficient operations. The HDM algorithm shall use a combination of counts of the number of headers added to the file, the elapsed time since a header was added and the elapsed time since the PHI was initiated to determine when to initiate the PHI. The HDM shall initiate the Packet Header Interface whenever the condition  $(A+B+C+D)$  is true, where "+" is the logical "OR."

A = 1 when communications with the staging area are terminated as determined by exchange of supervisory packets. The SAI microprocessor can communicate condition to VAX 1 via fiber optic data bus.

B = 1 when allocated file space is more than fifty percent utilized. The Header Data Manager maintains an internal map of the next available location on the auxiliary storage file.

C = 1 when the elapsed time since the last initiation of the Packet Header Interface exceeds "T." Initial "T" shall be 120 seconds. This will permit the receipt of approximately five Thematic Mapper Images in real time if there were no interfering communications. It will allow for the receipt of one image with a twenty percent duty cycle on communication.

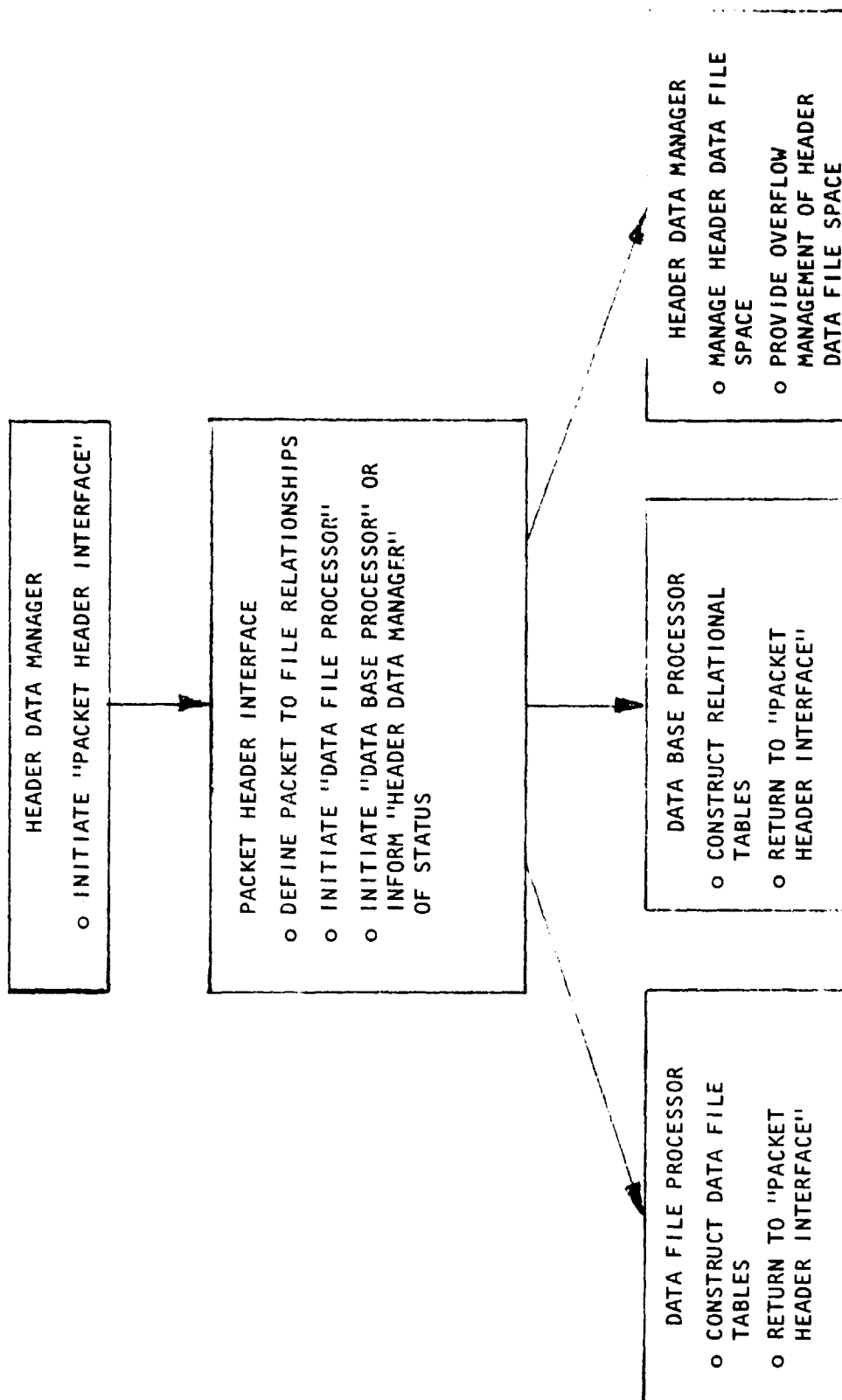


Figure 2-6. Process Flow for Header Data

D = 1 when the count of the received packets exceeds "N" since the last initiation of the Packet Header Interface. Initial "N" shall be 600.

#### 2.4.5.2 Packet Header Interface File Processor Management

Upon initiation by the HDM, the PHI shall process the headers to establish the relationship of IP's to files. This function shall be performed in an automatic mode. The PHI shall use the appropriate routines and tables of the Data File Processor (DFP). Pre-established tables shall define the packet to file relationships according to source ID, mission ID, source sequence count, and information in the secondary headers.

After the Packet Header Interface completes the task of establishing the packet to file relationships, it shall initiate the Data File Processor to perform the function of updating the data file tables. Upon completion of the data file processing, control shall return to the Packet Header Interface. Logic within the PHI shall permit the option of either directing the process to the Data Base Processor for the construction of relational tables or to the Header Data Manager for the purging, compression, and management of the header data file space on auxiliary storage.

The entire process of defining the packet to file relationships, constructing data file tables, and constructing relational tables shall be capable of being interrupted and restarted successively. The management of the header data file space by the Header Data Management, including the overflow to the AMM and subsequent reconstruction of the data on the auxiliary storage shall be transparent to the Packet Header Processor, the Data File Processor, and the Data Base Processor.

#### 2.4.5.3 Packet Header Interface Data Base Processor Management

When the data file tables have been established and the header data is in auxiliary storage, either as a result of direct placement or retrieval from AMM, the HDM shall call the PHI to establish relational tables. The relational tables may be established automatically by the system using the PHI or manually by the user.

In addition to the automatic mode just described, the PHI shall operate in a manual mode with human operator intervention. In the manual mode, the operator shall have the ability to establish relations according to logical conditions.

The PHI shall be capable of being restarted when performing data base processor management. Header data for any sets of packets or files may be retrieved from the AMM for the construction of additional relations. This will predominantly be performed manually, but the modification and subsequent automatic mode operation shall not be precluded.

#### 2.4.5.4 Priority of HDM Functions

Priority of functions performed by the HDM shall be assigned to ensure that no packets are lost in the AMM. In keeping with this objective, the functions of the HDM are listed in descending priority.

1. Maintain adequate file space on auxiliary storage for header data.
2. Initiate the PHI to perform Data File Processor Table Updates.
3. Initiate the PHI to perform automatic Data Base Processor Table Updates.
4. Initiate the PHI to perform manual Data Base Processor Table Updates.

#### 2.5 DATA BUS TRANSFERS

All transfers of data on the data bus shall be initiated by VAX II. VAX II shall insure that only one port is transmitting at a time. As many ports as necessary may receive simultaneously. VAX II shall initiate transfers at the packet level.

The data packet transfer shall precede without additional intervention by VAX II. All transfers shall be blocked at a maximum block size of 2048 bits. Each block transfer shall be asynchronous and under the control of a Master Controller (MC). Synchronization, control, and a second synchronization block shall precede each data block. Transfers of both control and data blocks shall be synchronous from the transmitting port to the receiving ports without intervention of the Master Controller.



### 2.5.1 BUS CONTROL CONCEPT

The bus control concept is intended to accommodate a combined data rate of 200 megabits per second. There shall be two levels of time division multiplexing. The bus shall provide for two classes of data transfers; one from the SAI to the AMM termed class I and the other termed class II that may be between any other ports. Class I data transfers may also be received by other ports. The two classes of data transfers shall be time division multiplexed at the bit level so the electronics for each will operate at only half the speed of the composite bus transfer rate. Data transfers will be synchronous within specified time slots. The timing cycle is illustrated in Figure 2-7.

SYNCH	CONTROL	SYNCH	DATA 2048 BITS
4.0 microseconds	1.84 microseconds	4.0 microseconds	20.48 microseconds

Figure 2-7. Data Bus Time Slots

All ports shall enable automatically under control of their internal clocks during each 4.0 microsecond SYNCH time period to permit synchronization. The SYNCH time shall be sufficient to allow settling of hardwire control lines between the master controller and the bus ports.

The data field shall be 20.48 microseconds long which will correspond to 2048 bits at 100 megabit per second which is the electronic transfer rate. Within this time period, both class I and class II data shall exist. This shall be effected by a 250 megabit compatible pulse duration, even though the repetition rate for any transmitter or receiver shall be 100 megabit. This is described in Section 3.0.

The control time shall be 1.84 microseconds which will permit the transfer of 184 bits of control information. The control information is described in Figure 2-8.

Function Code 16 Bits	Operand 120 Bits	Auxiliary Operand 40 Bits	CRC 8 Bits
--------------------------	------------------	---------------------------	---------------

Figure 2-8. Data Bus Control Time Slot

The Master Controller shall interface to VAX 11 via the Unibus. The Master Controller shall format the control words or output them from VAX memory, manage bus access, and manage the block transfers required for packet transfers.

The microprocessor or functional replacement for some terminal classes, shall have at least four TTL compatible hardwired lines to the Master Controller. They are described below.

- 1) Poll: This signal directs the ports to become active. The Master Controller shall simultaneously raise this line to each of the designated transmitting ports and the receiving ports when a transfer is to be affected. This shall occur during the synchronization time on the bus so transit time difference will not be critical. It shall remain raised throughout the data transfer.
- 2) Inhibit: The signal shall be raised by any port that is in the process of transferring data in or out of its device buffer. The Master Controller shall not raise a poll line to any port with a raised inhibit. Any port can raise its inhibit whenever it is unavailable for any reason. A permissible reason might be the unavailability of the interfacing device, as determined by the microprocessor. However, such conditions could also be reported in detail using the appropriate function codes and protocols.
- 3) Acknowledge: This line shall be raised by the receiving port upon completion of a block transfer. It shall be a signal to the Master Controller that the CRC was acceptable and the data was transferred to the 2K buffer.
- 4) Interrupt: This shall be a line that any port can raise to interrupt the Master Controller to initiate communication. Upon receipt of an interrupt, the Master Controller shall successively read its interrupt register and poll the interrupting port to determine the nature of the data transfer.

#### 2.5.2 CONTROL TIME

The control time shall be designated according to Figure 2-8. The sixteen bits of function code permit flexibility and expansion capability of the concept. Functions are described in paragraph 2.10.

The operand shall accommodate the following packet identification. It is made up as shown in Table 2-2.

Table 2-2. Packet Identification

Title	Bits
DBMS Header (Source, block count, multiple bit)	8
Identification Source ID or equivalent	8
Identification Mission ID or equivalent	8
Source Sequence Count	32
Starting Location in Packet (bytes)	32
Total Transfer (bytes)	32
Total	120

For functions other than transfer of blocks of data, the operand data may be used differently. To allow for expansion, an additional 40 bits of auxiliary operand shall be reserved. The control word is completed with 8 bits of CRC data.

### 2.5.3 TYPICAL CLASS II PACKET TRANSFER SCENARIO

For the Class II transfer of a given packet of data, the Master Controller shall accept control words from VAX I; interpret the control words; set up the transmitting port and the receiving port(s); instruct the respective ports on their duties and poll them for their preparedness, which may involve interrupts; schedule the ports for the entire packet transfer without interference; and then monitor each block transfer.

When all ports are ready, as determined by the drop of the inhibit from the designated ports, the Master Controller shall raise the appropriate poll lines during the synch period immediately preceding the data transfer time slot. The success of the transfer shall be determined by the Master Controller via the acknowledge lines.

Control words shall be transferred at the appropriate times in a similar manner. Several microseconds may be required between block transfers to permit a local microprocessor to either interpret the data if it were a control word, or to complete the transfer to the device on the other side of the

Interface. The design shall permit the substitution of an external computer for the internal microprocessor controlled device interface logic for high data rate external devices. A multiplexed port will also increase overall bus utilization.

## 2.6 STAGING AREA INTERFACE

The SAI performs both MDTS and DBMS functions. The SAI DBMS functions were described in paragraphs 2.3 and 2.4. The SAI MDTS functions include X25 frame checking and level 4 protocol.

### 2.6.1 SPECIAL CONDITIONS

The functions described in the following paragraphs shall apply to the initial MDTS interface. While all internal DBMS system functions are specified to accommodate a 50 megabit per second data transfer with growth potential for 100 megabits per second, the initial interface shall be implemented according to X25 protocol at 56 kilobits per second. The SAI shall be sufficiently modular to minimize the impact of a future change in protocol and data rates.

### 2.6.2 LEVEL 4 PROTOCOL

The protocol for establishing communications between the MDTS/Staging Area and the DBMS shall follow X25 standards. There shall be supervisory and information frames as defined in paragraph 2.3.2 of the X.25 standard. Supervisory packets shall be used to maintain data integrity. Additionally, the control field for each information frame shall be used according to the conditions of the following paragraphs.

#### 2.6.2.1 Flag and Transparency

The start or end of a frame shall be indicated by a flag which is 01111110. To prevent the synthesis of a flag code by any of the data including the X.25 header and frame check bits, additional zeros called "transparency" bits are inserted every time five successive ones are encountered. The SAI shall remove a zero every time five successive ones and a zero are encountered. See paragraph 2.2.6 of X.25 standard.

#### 2.6.2.2 Address

The first 8 bits following the receipt of a flag are address bits that are used by the network for routing. Only valid packets are assumed at the SAI. It shall not be required of the SAI to perform an address check since a bad address will only occur during a network malfunction which will be detected by other error detection methods.

#### 2.6.2.3 Control Field

The control field, bits 9 through 16, shall be decoded. Bit 9 will signify either an information frame (0) or supervisory (1) packet. X.25 L3 packet bits shall be decoded and the multiple bit shall be used to identify when additional packets are expected to complete the IP.

#### 2.6.2.4 Error Detection

The SAI shall perform three redundant error detection functions. At X25 level 2, each frame sequence shall be checked to insure that the frames are in order. At X25 level 3, the multiple bit shall be checked to insure no frames are lost. At the frame level, a 16-bit frame check sequence (FSC) shall be computed and compared with the FCS transmitted with the frame.

#### 2.6.2.5 Supervisory Packets

At the completion of each error check, the appropriate supervisory packets shall be generated and returned to indicate either frame rejection, abort, idle channel, or acknowledge. Both the frame rejection and acknowledge supervisory packets shall include the frame sequence count. An acknowledge is sent if the CRC, a frame sequence, and the pole final (P/F) bit in the control field are true. Satisfaction of all but the P/F bit shall result in no return supervisory packet. Failure of any other condition shall result in an immediate transmission of a frame reject supervisory frame.

#### 2.6.3 TYPICAL SCENARIO OF RECEIPT OF DATA

Data shall be received according to X.25 packets (L3) and frames (L2). Multiple frames may be required for a packet, as indicated by the multiple bit in the X.25 L3 field. L3 packets shall correspond to instrument packets and shall

range from a minimum of 256 bits to a maximum of 8,388,577 bits. L2 frames shall be a minimum of 48 bits (supervisory) and a maximum of 2048 bits.

Each frame received shall be sequentially routed to one of 8 buffers. Additionally, the headers, both primary and secondary, shall be stored in a header buffer. The secondary header length is defined in the fixed length primary header.

A CRC, as well as frame sequence number, shall be checked on each frame. When errors are detected, the frame shall be aborted and the proper supervisory packet shall be returned.

The proper supervisory packet acknowledging the receipt of a good frame shall be returned when the checks are valid and the poll bit is set in the control field of the frame. Current plans are for the poll bit to be set on the initial and final frame of each packet as well as every fourth frame of multi-frame packets. The limitations of the frame sequence count to modulo 8 and the planned protocol requires the buffering of 8 frames of multiple frame IP's prior to their transfer to the AMM.

When a frame is rejected, all frames subsequent to the last acknowledged frame shall be retransmitted. Data frames shall be retained at the interface until either a new packet is received or until the frame sequence count repeats. This is to permit a retransmission of all frames back to, but not including, the last acknowledged frame. Failure to adhere to this provision could result in excess and non-contiguous frames being archived with a resulting excess overhead during data retrieval.

## 2.7 DATA RETRIEVAL FROM AMM

The AMM shall provide all the necessary internal packet management to retrieve data at the packet level or at a specified byte count within the packet. They shall be identified by 120 bits of address information as was shown on Table 2-2. Fifty-six bits shall uniquely define the packet, 32 bits the starting location within the packet and 32 bits shall identify the total number of bytes to be transferred.

### 2.7.1 IDENTIFICATION OF PACKETS

The IDBMS, through the execution of the DBP will identify a file or a part of a file that contains the desired data. The IDBMS will execute the DFP which will identify packets comprising the desired file. The DFP will identify a packet or a sequence of packets to the CMS which through a series of calls to the AMM driver and the Data Bus driver will initiate the retrieval of the data.

Processors called by the DFP shall construct a table of packet identification data. This table shall contain entries of 120 bits each. The AMM driver shall set up a series of calls to the Data Bus Driver (DBD) to command the retrieval of the desired data from the AMM.

### 2.7.2 TYPICAL SEQUENCE OF EVENTS IN DATA RETRIEVAL

The retrieval of data from the AMM would be initiated by the need of some user or process that would determine the destination of the data packets. This destination address is located in an interface table within the DBD. Such a sequence is described in paragraph 2.8. A detailed flow of the major software modules involved is presented in paragraph 4.8. For this scenario, it is assumed that the proper destination data is available to the DBD. The proper sequence of data, including commands, shall be constructed in VAX II and passed to the master controller. The maximum intervention of the VAX during a file transfer shall be once per packet. However, for multipacket files, fewer interventions shall not be prohibited.

#### 2.7.2.1 Master Controller Function

The MC shall place the appropriate command data on the data bus to direct the AMM to retrieve the desired data, to set up the receiver ports, and to initiate the transfers. Once a packet transfer is started, the MC shall not permit any other transfers to take place involving the designated ports. The MC shall direct the transfer of each block of data in a contiguous manner until the entire packet has been transferred. The time between block transfers shall be asynchronously controlled by the interfacing port devices.

#### 2.7.2.2 Look Ahead

A look ahead function is to be provided for as a future option for multi-packet files. With this option, appropriate retrieval commands may be issued to the AMM to enable it to retrieve all the packets comprising a desired file or files. This feature is advantageous for future larger archives that may require several seconds to retrieve off-line data. Each packet transfer would still be initiated by the MC.

#### 2.7.2.3 Retrieval Command

The following sequence of events shall transpire in a retrieval of a packet of data:

1. MC checks AMM Data Bus Port (DBP) inhibit line.
2. If the AMM DBP inhibit line is low, the MC loads port buffer with proper command words.
3. MC rechecks AMM DBP inhibit line.
4. If AMM inhibit is low, MC raises AMM poll line during command time.
5. Command words are received at AMM port.
6. AMM port raises inhibit while local port controller interprets data.
7. If CRC checks, AMM port raises acknowledge line to signify receipt of command.
8. MC performs other control functions.
9. AMM port controller interprets commands and engages in protocol exchange and data transfers (32 bit words) to AMM.

The AMM then performs the function of retrieving the required data from archive and staging it for transfer in a sequence of 2K blocks. As soon as the AMM port buffer is free to accept additional commands, the AMM DBP inhibit line is dropped.

#### 2.7.2.4 Transfer Setup

While the AMM is recovering and staging data for transfer, the MC shall continue to handle other DB transfers. It shall also set up the proper receiving ports according to the data provided by the Data Bus Traffic Control Processor.



For purposes of this discussion, assume the packet of data is to be transferred to User Terminal (UT)1. The following sequence of events shall transpire:

1. MC checks UT1 inhibit.
2. If inhibit low, MC loads port buffer with proper command words (get ready to receive data).
3. MC rechecks UT1 inhibit.
4. If inhibit low, MC raises UT1 poll line during command time.
5. Command words are received at UT1.
6. UT1 raises inhibit while local port controller interprets data.
7. If CRC checks, UT1 port raises acknowledge line to signify receipt of command.
8. MC performs other control functions and UT1 port prepares to receive packet.

#### 2.7.2.5 Data Block Transfer

While both the AMM and the UT1 ports are preparing for their respective functions at the time of transfer, their respective inhibit lines are raised. The AMM stages the entire packet. The AMM port controller loads the port buffer with the first block of data. As soon as each port is ready, it drops its inhibit line. The MC senses these inhibits and at the first opportunity after both the transmitting port and all the stipulated receiving ports have lowered their inhibit lines and no other data transfers are taking place on the bus, the MC shall raise the poll lines to all the affected ports during data time.

The entire 2K, or less for a small packet, of data is transferred to the designated ports, UT1, during the 20.48 microseconds of class II data time.

The procedures are repeated at the receiving ports. Inhibits are raised, CRC's checked, acknowledges raised, and data transferred out of the buffer under control of the local port controller. The transfer shall take place according to the device interface protocol for that particular port. The MC shall raise the acknowledge line to the transmitting port to signify the completion of a valid block transfer.

The AMM controller shall repeat the sequence of loading its buffer and lowering the inhibit when it is ready to affect another block transfer. The MC shall repeat the procedure of raising the appropriate poll line when the necessary conditions are satisfied. This sequence of events shall continue until the complete packet is transferred. Both the transmitting port controller and the MC shall maintain cognizance over the number of blocks required to complete the transfer. Upon the completion of the last block transfer, the transmitting port shall not raise the inhibit because it will not be loading its buffer. It shall raise the acknowledge line. The MC having its own count of the number of blocks expected shall check for the acknowledge. The MC shall then continue with its other functions.

### 2.7.3 ERROR CONDITIONS

A sequence of error detection and recovery events shall be implemented. Failure of a CRC shall result in a failure of an acknowledge to the MC and a subsequent failure of the acknowledge from the MC to the transmitting port. When the MC fails to receive an acknowledge, it shall immediately cause the transmission to be repeated by raising the appropriate poll lines during the next data or command time according to the failed message. The system shall provide for a flexible number of retransmission attempts that can be easily altered by maintenance procedures.

A system of error communications and contingency command words shall be provided to minimize irretrievable data loss due to sustained data transfer failure.

### 2.7.4 TIMING

Timing of each data transfer shall be synchronized by a synch code immediately preceding each control and data time. Since the synchronization code is transferred over an identical data path as the data, difference in transit times to different receiver ports will not be critical. All hardware control signal state changes shall occur during the synchronization times so the system will be in steady state when either a control or data word is to be transferred.

## 2.8 TYPICAL SCENARIO OF USER OPERATION

The main objective of the DBMS is to provide a friendly environment for users to obtain access to space data. This requires both a facile retrieval and a facile identification of the data, its attributes, and its location. The scenario of this section describes the initiation of a data identification from a user terminal, the interaction of a user with the Packet Header Interface and the initiation of the data retrieval. The major hardware and software involved in this scenario are described in paragraph 4.8.

### 2.8.1 USER TERMINAL FUNCTION

Although the DBMS shall function similarly for a number of different user terminal classes, including special purpose processors, the scenario described shall pertain to an image CRT device with a keyboard and trackball input. The detailed mechanics of the interface between the user and the device shall be assumed. Similarly, the necessary protocol to transfer the commands and the return data across the interface between the terminal controller and the data bus port is assumed. The port controller shall have sufficient capability to interpret the commands and to assemble the appropriate command words for communication on the data bus. The data bus port shall have the capability to treat data and commands originating in the terminal controller as data and pass it through to software processors in the VAX. Such data shall be held distinct from specific control words and shall be transferred during the data transfer time.

### 2.8.2 INTERRUPT

Each data bus port shall have the capability to interrupt the MC. When a port wishes to initiate communication, it shall initiate the following sequence of events:

1. Raise inhibit.
2. Load buffer with proper command or data.
3. Raise interrupt and lower inhibit.

When the MC senses an interrupt, it shall check its interrupt buffer to determine the interrupting port. It shall then check for a raised inhibit and then

raise the poll line of that port during the command time. It shall also raise the poll on its own port to receive the command word from the terminal it wishes to initiate a data transfer. The appropriate command words shall signify the nature of the desired communication. The sequence of acknowledge, inhibits and interpretation by the local controller shall be followed similar to the bus transfers described in paragraphs 2.7.2.3 and 2.7.2.4.

Once the MC receives an interrupt, it shall follow the above procedure until each interrupting port has been polled. All received command words shall be processed to the point that cognizance of desired transfers are maintained.

### 2.8.3 QUERY

Toward the goal of providing a friendly environment for the user of space data, the DBMS shall provide assistance in both the location and identification of the desired data and in the use of the system. The DBMS shall provide, with the appropriate function codes, the ability to pass through user commands to the IDBMS. The DBMS shall incorporate the IDBMS and its user interactive capability in its entirety. The following paragraphs illustrate how that capability could be used by a user signing on the system and desiring some data from AMM. The software flow for this scenario is detailed in Section 4.

The success of the scenario described below is dependent upon the previous construction of the necessary tables in the IDBMS. It is intended to illustrate how the system might be used and shall not be considered a specific requirement for the data described except that the system shall be compatible with the IDBMS.

#### 2.8.3.1 Data Desired

The user wants to obtain a recent image of visible data of wheat lands in the Cheyenne River Valley of South Dakota. He knows the area of interest is near Pedro. The time is early spring and he is concerned about flood conditions, snow melt, ice jams, and wheat field conditions.

### 2.8.3.2 User Activity

The user signs on the terminal according to established procedures. He asks for help in using the system. He is prompted in the necessary query language. Each of these interactions is handled in turn by the data bus. VAX II instructs the MC of the need for interactive user terminal communication. As a function of the IDBMS interaction technique, VAX II may poll the terminal for periodic inputs. This is not a requirement of the DBMS, but the DBMS shall accommodate this feature if implemented in the user software.

The user identifies the category of data of interest, the time frame of interest and the general geographic region. The IDBMS asks if he wants a specific area. The user says "yes," Pedro, South Dakota and the Cheyenne River. The IDBMS proceeds through its relational tables to identify the area around  $-102^{\circ}$  west longitude,  $44.5^{\circ}$  N latitude. It also identifies an image taken April 14 at 9:02 in the morning just 2 days ago. The user confirms that he would like to see that image.

### 2.8.4 DATA BASE PROCESSOR ACTIVITY

During the activity described above, the Packet Header Interface is performing various logical operations, table searches, and accesses to the auxiliary storage for the necessary tables. It calls the drivers and passes appropriate messages to other processes. A user interactive processor, which in turn uses the DBP to provide relational tables for its activity, is called to assist in the user prompting. Once the particular file is identified, the required packets are identified and retrieved from the AMM in a manner similar to that described in paragraph 2.7. In this example, only a small area is desired and since the data is recent, it has not been processed. Thus, the data is brought into the auxiliary storage and additional user interaction and processing are required.

### 2.8.5 IMAGE UTILITIES

The DBMS shall provide a minimal amount of image and other sensor data processing utilities. They will be primarily formatting and others necessary to select portions of data packets. The software required to display the data

on the User Terminal shall also be provided. Additionally, there shall be provisions for the operation, in a low priority mode, of user supplied information extraction software. This latter requirement is for demonstration purposes only.

#### 2.8.5.1 User Image Processing Assistance

Since the data is still unprocessed, the first step is for the DBMS to determine what processing is required. A user assistance processor shall be called that will use the DBP to locate a processed image of the desired geographical area. This image shall be retrieved and presented directly to the UT for display. It shall be already stored in the AMM in a format directly compatible with the UT display. The controller in the UT port shall have the capability to remove the DBMS header data prior to transferring the data from the port buffer to the display controller. All the necessary display controller command words shall also be included in the archived data so no additional processing is required.

The displayed image shall provide the user with a map of the area. He can then identify using the track ball, the area of interest. This information shall be transferred to the user assistance processor. There it shall be interpreted to set up the proper editing routines.

Additional information on the desired processing shall be obtained from the user. The user information shall define projections, geometric corrections, resampling algorithm, and the desired contrast stretching. Default values shall be provided internally for the unsophisticated user that either does not know or does not care.

#### 2.8.5.2 Image Format Processing

The data temporarily stored in auxiliary storage shall be transferred to either VAX I or VAX II for processing by the Image Format Processor (IFP). Because of memory size limitations, it may be necessary to bring the data in a small portion at a time. It may also be necessary to repeatedly bring in the same data since the IP's for image data are likely to contain data continuously stretching from one edge of a scene to another. Each IP may

need to be brought into the VAX, the portion of the scene desired removed and returned to auxiliary storage, and the process repeated until a packet of manageable size covering the desired area is assembled. That temporary packet would then be brought in and processed.

After the desired image is processed, it must then be formatted suitably for display. The necessary control words for the display controller must be inserted by the Image Format Processor. When the data is ready for display, the Data Bus Traffic Control Processor will be notified to transfer it to the User Terminal port.

#### 2.8.6 ADDITIONAL FUNCTIONS FOR THE USER

The DBMS shall retain all intermediate data either on the auxiliary storage or in archive until such time as the Header Data Manager purges the temporary files. This may happen as a result of either an elapsed time without activity or upon explicit commands from the user.

The DBMS shall be able to provide the data to other ports for processing on the users' equipment. For demonstration purposes, the data shall be made available to user provided information extraction processes that will operate in VAX II.

#### 2.9 DBMS PACKETS

A significant part of the data archived in the DBMS will be processed instrument data and ancillary data. Some of the data will be in display compatible format. This data is referenced as DBMS packets. There shall be different classes of DBMS packets that shall be recognized by the 2nd through the 7th bits in the DBMS header. DBMS packets shall be distinguished from instrument packets by the first bit, which shall be a (0) for DBMS packets. For a description of instrument packets, see paragraph 2.3.3.1.

##### 2.9.1 DATA BLOCKS

All data transfers on the data bus shall be in blocks of 2048 bits or less. The first 8 bits of each block shall contain the block header described above.

The last 8 bits shall contain a block CRC. Both the block header and the CRC shall be retained with the data when it is archived.

A typical DBMS packet is shown on Figure 2-9. The first block contains 32\* bits of unique identification code. These codes shall be assigned by software in the CMS and within blocks by software in the IDBMS. The CMS shall utilize a combination of algorithms and table look ups to maintain a unique set of identification codes.

The maximum permissible DBMS packet lengths shall be 8,388,577 bits to be consistent with instrument packets. Each DBMS packet shall have a 16-bit packet parity as the final data entry prior to the block CRT.

#### 2.9.2 DBMS PACKET TYPES

The DBMS Packet Format can accommodate 64 types of packets. This is in excess of the currently identified need. The minimum number of types of DBMS packets that shall be implemented are identified in Table 2-3.

The packet types are planned to provide a rapid identification and cross check for the major categories of archived data. These categories are:

1. Software
2. Ancillary Data
3. Processed Data
4. Data suitable for direct transfer to devices
5. Data used internally by DBMS

#### 2.10 DATA BUS FUNCTION CODES

A set of function codes shall be established to provide for consistent and flexible control of the present and expanded DBMS. As a minimum, the functions listed in Table 2-4 shall be accommodated within the 16 bits function code allocated to the control words.

---

\* May be optionally increased by 16 bits.



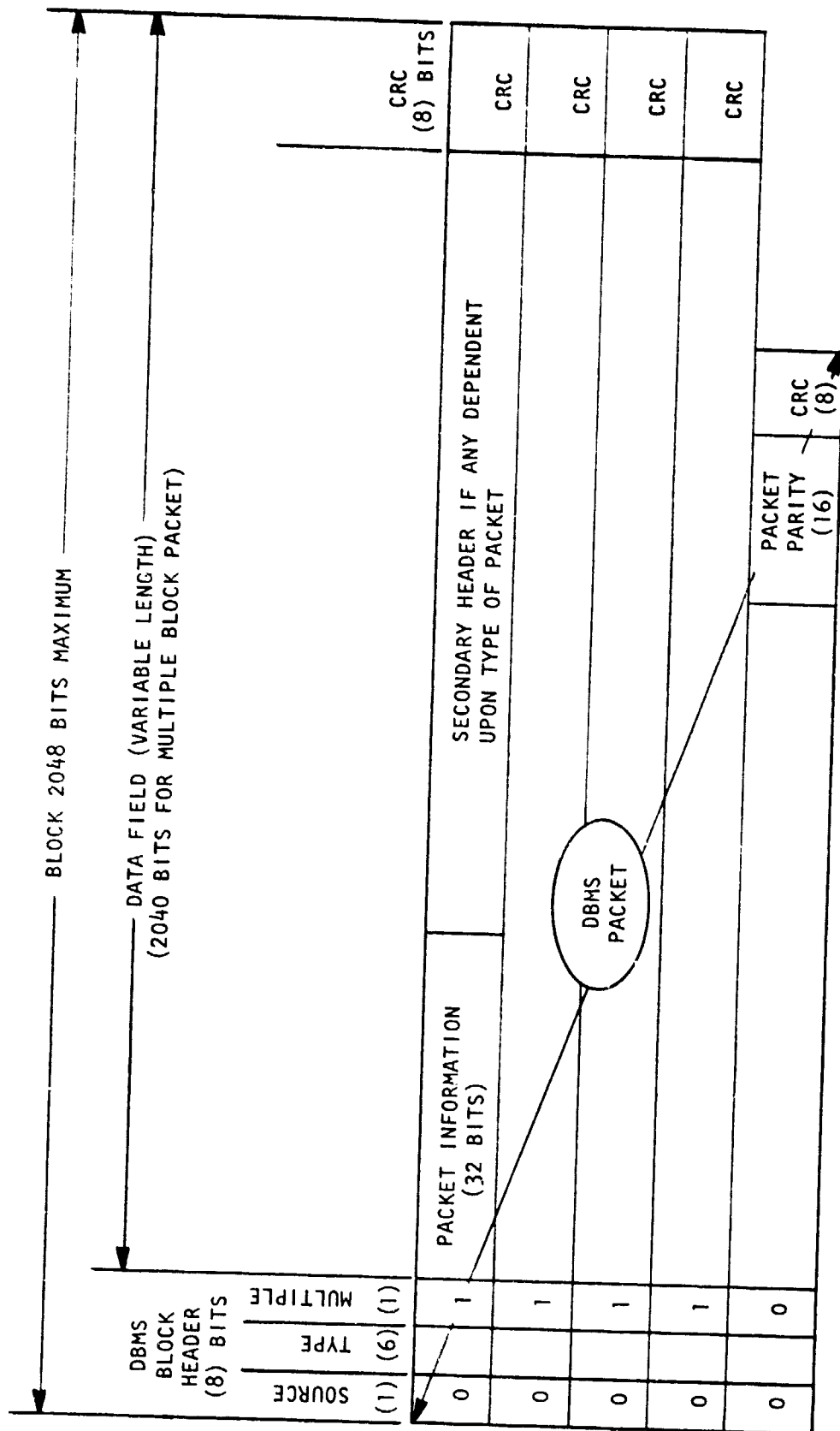


Figure 2-9. DBMS Packet Format

Table 2-3. DBMS Packet Types

CODE	TYPE
001001	Free Form Data with No Secondary Header
001010	Data with descriptive Secondary Header of Fixed Format
001100	Data with descriptive Secondary Header of Fixed Format
100010	Display data with embedded control Type 1 Display
100100	Display data with embedded control Type 2 Display
100110	Display data with embedded control Type 3 Display
110010	Relational Table
110100	DBMS Management Data
110110	Software Processes
111010	Ancillary Data
111100	IDBMS
111101	IDBMS
111110	IDBMS
	To Be Assigned

Table 2-4. Bus Function Codes

Prepare to transmit packet from staging area  
 Prepare to receive packet from staging area  
 Prepare to transmit packet from DBMS  
 Prepare to receive packet from DBMS  
 Decode operand for format information  
 Pass through operand for device control  
 Decode operand for device control  
 Respond to polling query  
 Acknowledge receipt of packet  
 Request for retransmission  
 Prepare to transmit block of data  
 Prepare to receive block of data  
 Abort packet transfer  
 No operation  
 Return test data  
 Test data returned  
 Establish user terminal communication

## SECTION 3

### HARDWARE SPECIFICATION

The DBMS comprises the following subsystems and major hardware elements:

1. Integrated Data Base Computer (VAX-1)
2. Configuration Management Computer (VAX-11)
3. Archival Mass Memory
4. Auxiliary Storage
5. Staging Area Interface
6. User Terminals
7. Data Bus

The DBMS shall be configured as presented in Figure 3-1. Protocol compatibility and interconnection of the DBMS is provided by the data bus element. All elements of the DBMS, other than the data bus are standard off-the-shelf hardware. The data bus consists of the data bus ports, cabling (fiber optics) and the star coupler (a transmissive fiber optics coupler). Although fiber optics buses have not been extensively implemented in fielded systems, all of the components required for its implementation are available from reputable companies. The data bus is a time division multiplexed system. The path between the staging interface and the Archival Mass Memory (AMM) shall be a dedicated channel. The paths between all other DBMS elements shall share a channel. This scheme permits data to be transferred from the staging interface to the AMM uninterrupted at a 100 MHz rate. Data transfers between the other ports are accomplished at a 100 MHz rate when enabled by the Master Controller.

Each element of the DBMS and its interfaces with the other elements of the DBMS will be described in the following paragraphs.

#### 3.1 INTEGRATED DATA BASE COMPUTER (VAX-1)

VAX-1 is an RP06/TU45-based VAX-11/780 system. A functional block diagram of VAX-1 is presented in Figure 3-2. This figure identifies each element of VAX-1 and its interfaces with other DBMS subsystems/components. The external interfaces

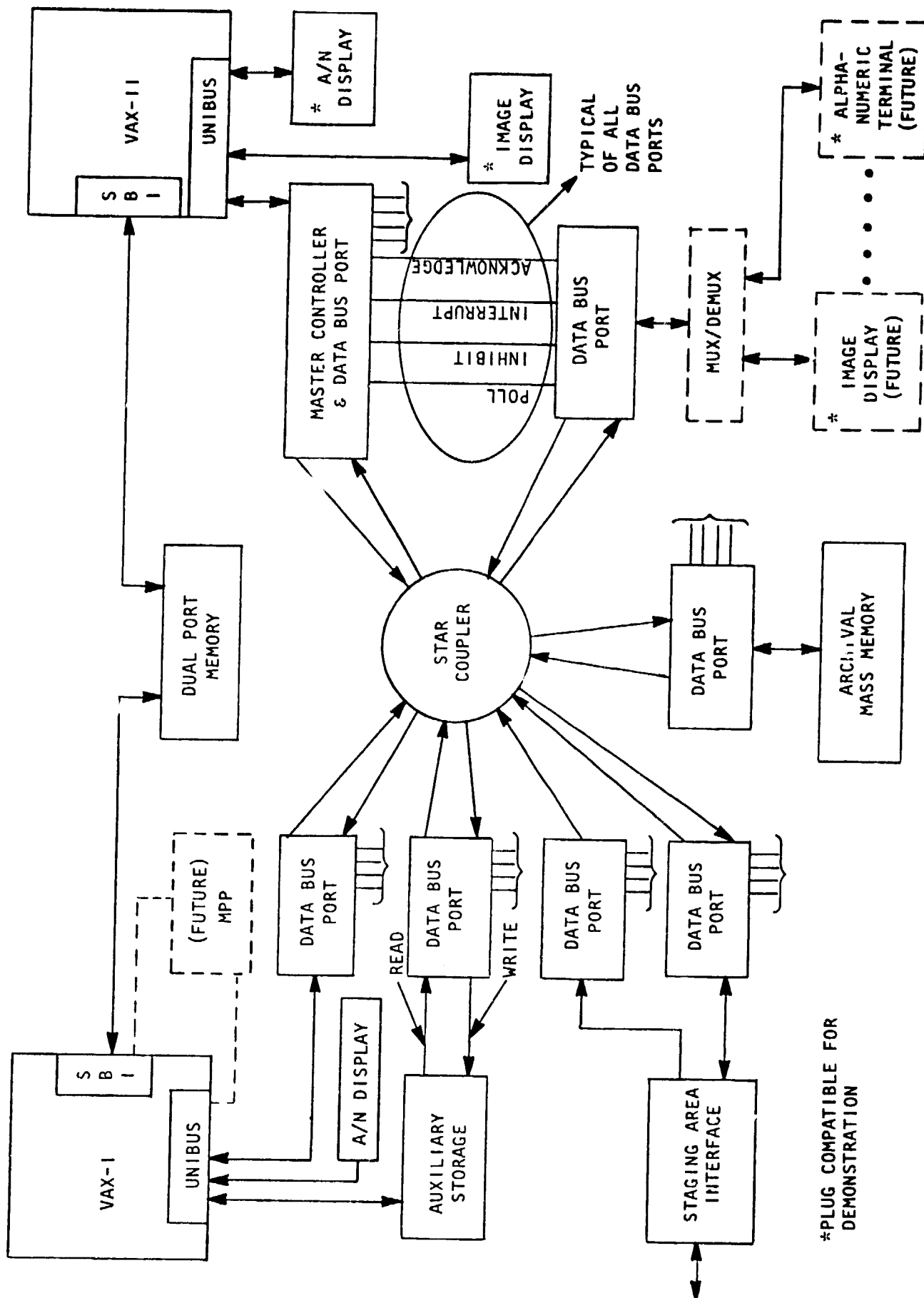


Figure 3-1. DBMS Hardware Configuration

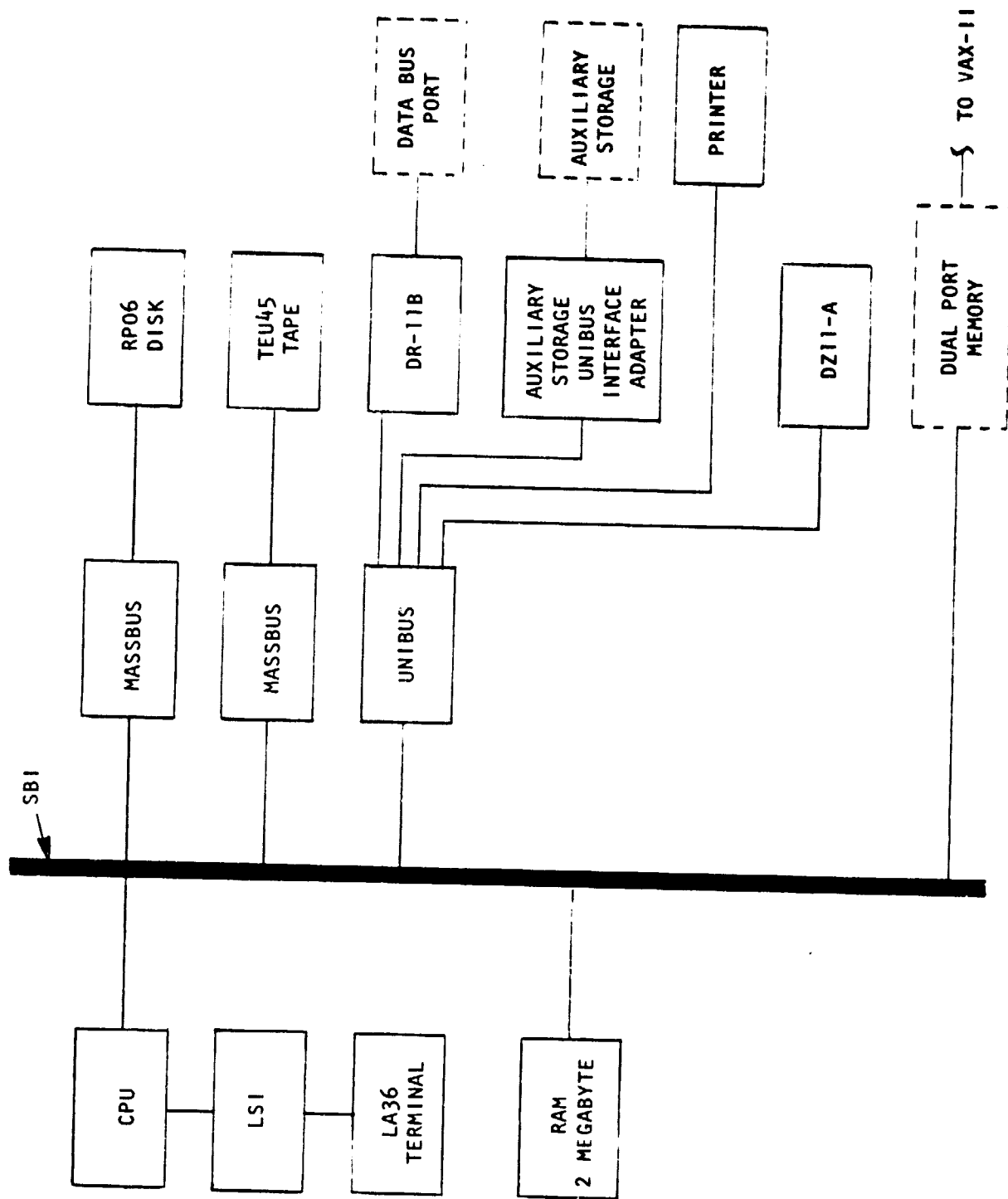


Figure 3-2. VAX-1 Functional Block Diagram

are identified by the dashed-line boxes. Each external device interface will be compatible with the associated (connecting) VAX-I interface (DR-11B, DZ11-A and the auxiliary storage UNIBUS interface adapter). All of the items in Figure 3-2 except the printer and the auxiliary storage UNIBUS interface adapter are provided as part of the RP06/TU45-based VAX-11/780 system. The GFE'd printer type is TBD. The auxiliary storage UNIBUS interface adapter is provided by the auxiliary storage subsystem supplier. The VAX-I computer system is GFE.

### 3.2 CONFIGURATION MANAGEMENT COMPUTER (VAX-11)

VAX-11 is an RP06/TE16 based VAX-11/780 system. The VAX-11 hardware configuration is presented in Figure 3-3. Its interfaces with other DBMS elements are identified by the dashed-line boxes.

VAX 11 has the primary function of managing the DBMS configuration and initiating data transfers via the data bus. The Master Controller data port controls the data transfer initiated by VAX-11. All DBMS elements interfacing with VAX-11 shall be compatible with the associated connecting DR-11B, DZ11-A, or UNIBUS interface as indicated in Figure 3-3.

### 3.3 ARCHIVAL MASS MEMORY

The AMM interface with the other DBMS elements is via a data bus port. The port provides the capability for simultaneous parallel data storage and retrieval. Data will be transferred in 32-bit words. The AMM provides the capability for simultaneously receiving and transmitting data at a 50 MBS. The AMM is GFE.

### 3.4 AUXILIARY STORAGE

The auxiliary storage subsystem will be an AMPEX Corporation Parallel Transfer Disk (PTD) system. The hardware configuration is presented in Figure 3-4. It interfaces directly with a VAX-I UNIBUS port and a data bus port. The data transfers to and from the auxiliary storage are specified via the UNIBUS interface. Data transfers (low speed) can also be accomplished via the UNIBUS interface. The high speed parallel transfers to and from the auxiliary storage

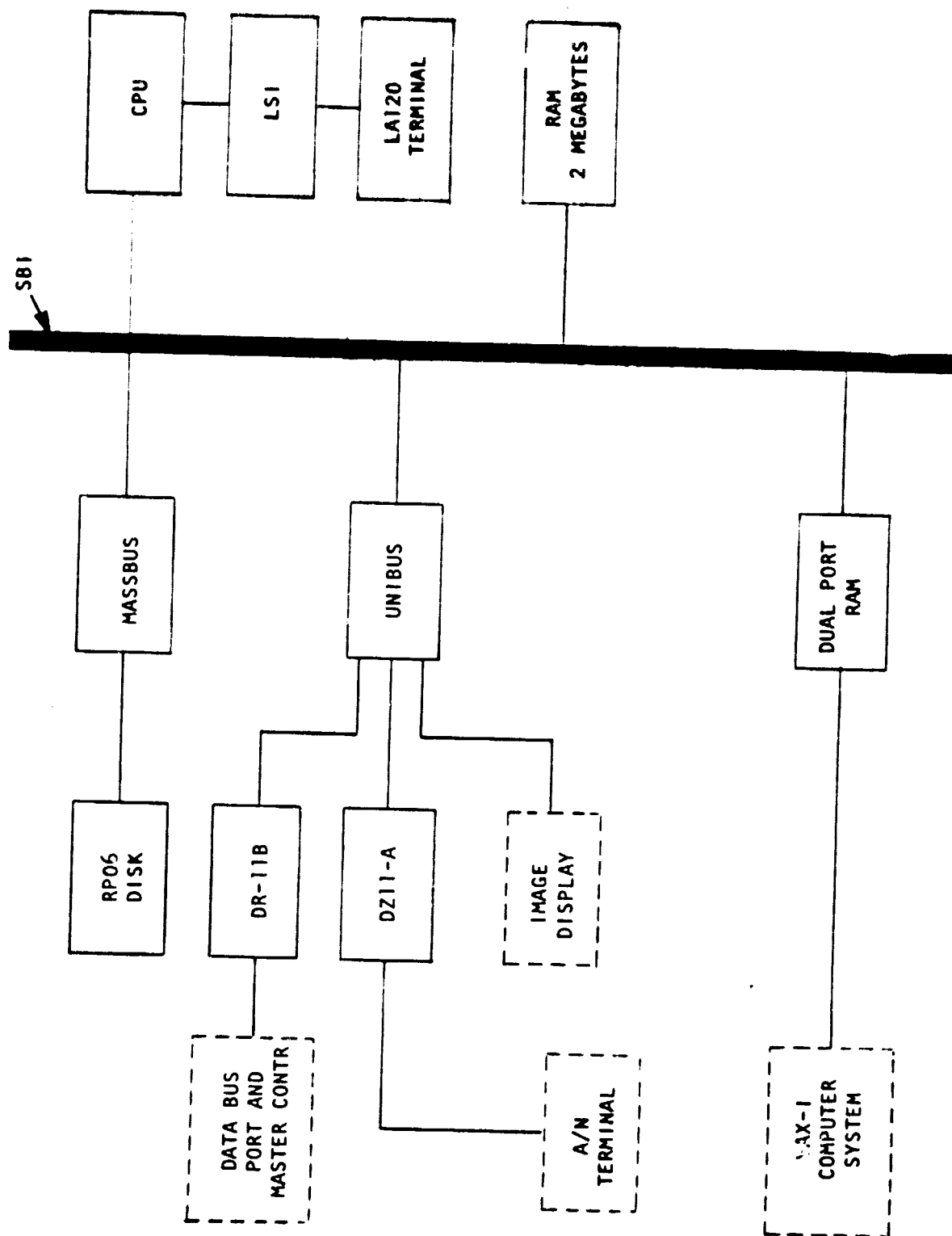


Figure 3-3. VAX-11 Hardware Configuration

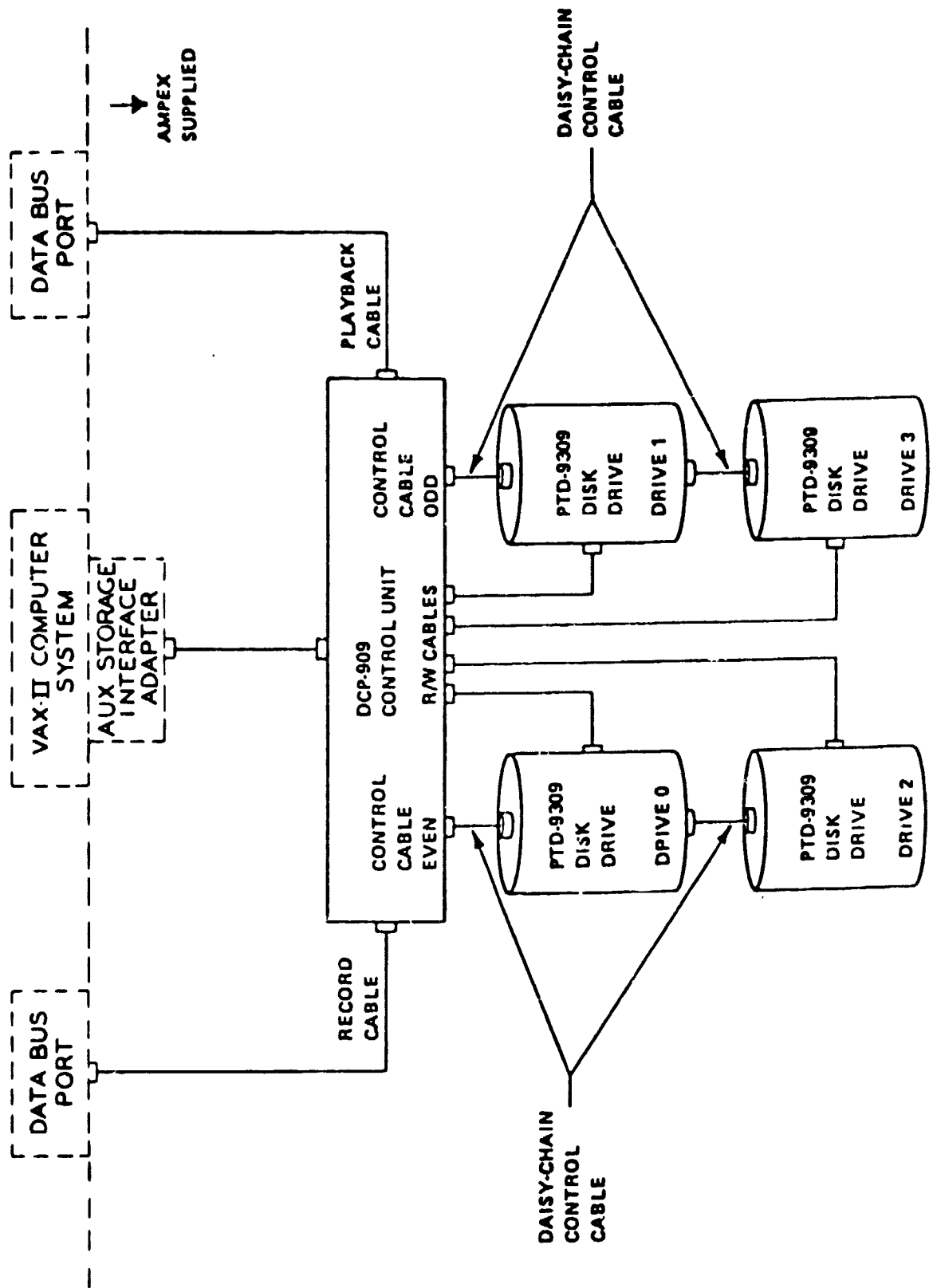


Figure 3-4. Auxiliary Storage Hardware Configuration



are accomplished via a data bus port. The PTD controller and disk drive are specified in AMPEX specifications 3309527-01 and 3308829-01, respectively. AMPEX also provides the auxiliary storage/UNIBUS interface adapter. This interface adapter will be physically located in VAX-1 and requires two card slots.

### 3.5 STAGING AREA INTERFACE HARDWARE

The Staging Area Interface (SAI) provides the interface between the DBMS and the principal sources of space data over external communication channels. The SAI shall be modular to minimize any impact of change in external channel protocols or data rate. There shall be three distinct modules which may share packaging and power supplies. Those modules that functionally perform the data bus Class I and Class II data interfaces shall operate at 50 MBS rate. These modules are specified in paragraphs 3.13 through 3.13.2. Hereafter, the staging area interface shall specifically refer to that end item performing the external communication interface.

A functional block diagram of this interface hardware is presented in Figure 3-5. It shall receive and transmit data serially with the staging area (a GSFC interface). This interface shall be compatible with the X.25 communication protocol. Salient features of this protocol were described in paragraphs 2.6 through 2.6.2.5.

### 3.6 USER TERMINALS

The DBMS shall include two types of display units, an alpha-numeric and an image display. These display units may interface with the DBMS via a data port and a multiplexer/demultiplexer unit or via a VAX-1 UNIBUS and DZ11-A interface (see Figure 3-1).

#### 3.6.1 ALPHA-NUMERIC TERMINALS

There shall be a minimum of two (2) monochromatic alpha-numeric CRT terminals with keyboards in DBMS. They shall be raster scanned with self-contained refresh memory and conform to the following specifications:

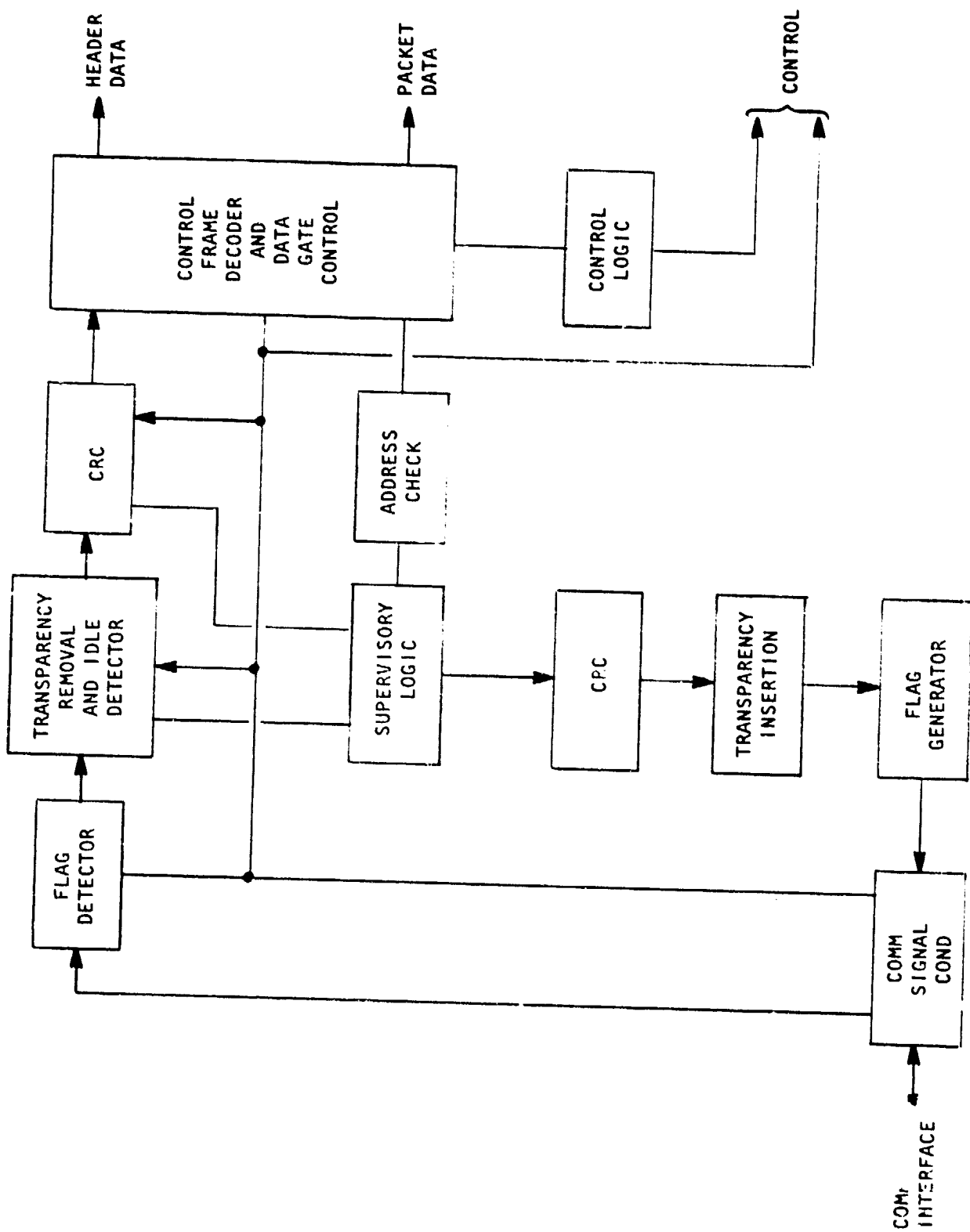


Figure 3-5. Staging Area Interface Hardware Functional Block Diagram

<u>Type</u>	<u>Raster Scan</u>
CRT Size	Minimum 12 inch measured diagonally
Display	25 lines by 80 characters
Character Size	Approximately 0.25" high by 0.1" wide
Character Type	7 x 8 dot matrix
Video Levels	Normal, blinking, half intensity
Character Generation	128 characters
Refresh Rate	60 Hz
Communication Interface	RS-232C
Transmission Rate	Selectable to 9600 bps
Communication Mode	Full duplex, half duplex
Erase Function	Erase from cursor to end of line Erase from cursor to end of memory Clear entire memory. Erase to end of field
Edit Mode	Local page and line edit, character insert and delete
Bell	Audible alarm which sounds at 72 position on line
Keyboard	Standard ANSI Configuration
Input Voltage	115V + 10% 60 Hz

### 3.6.2 IMAGE DISPLAY

There shall be an image display system containing a display controller and two independent three color CRT displays. The system shall receive, decode, scan-convert, and store computer generated alpha-numeric, graphic, and image data into a dual ported digital refresh memory system. It shall scan the stored picture at the television raster rate to produce the desired video signal. It shall have optional features to provide transformation and contrast stretching through the use of look up tables, zoom, windowing, blink, plot, bar chart generation, and filled polygons. Each work station shall have both a keyboard and a trackball input device. The controller shall interface to a DEC DR-11C in both programmed I/O and DMA modes. An additional RS232C interface shall be provided. The system shall conform to the following specifications:

Chassis	Include necessary power supply, card slots, backplane connections
Channels	Minimum of 6 to accommodate 2 independent RGB user work stations
Resolution	512 x 640 pixels per work station
Quantization Level	4 bit per channel
Overlay	Separate Alpha-Numeric annotation
Blink	Standard and reverse color
Keyboard	Standard 128 USASCII characters
Trackball	Trackball position with enter function button
Input Power	115 V + 10% 60 Hz
Expansion	Expandable to 4 work stations each with independent display
Interface	DEC DR11-C bidirectional. Operate PIO and DMA modes
Special Feature	Programmable color, intensity, blink assignment, and polygon filling
Serial Interface	RS232-C
Monitor	RGB high resolution
Monitor Size	Minimum 19" diagonal
Monitor Input Power	115 + 10 V 60 Hz
Monitor Input Signal	RGB video with composite synch on green
Monitor Input Connector	BNC
Video Bandwidth	Minimum 35 MHz
Operation	Non-interlace

### 3.7 DATA BUS

All DBMS data transfers shall take place on the fiber optic data bus. The fiber optic data bus shall consist of the fiber optic cable, a transmissive star, optical receivers and transmitters, and connectors. Every data bus

port shall have two fiber optic links, one for receiving and one for transmitting. All fiber optic links shall be connected to the transmissive star as shown in Figure 3-1.

### 3.7.1 DATA BUS STRUCTURE

The data bus is patterned after the IEEE 488 bus. Bus operation shall be handled by the Master Controller. The Master Controller will interface to the VAX 11 via a DR-11B UNIBUS port. The Master Controller shall format the control words or output them from VAX memory, maintain order of the bus transfers, and handle the block transfers required for packet transfers. The Master Controller along with the VAX interprets all requests for data transfers; identifies the data to be transferred, where it is, and which port will transmit; ascertains the receiving ports and their availability; sets up the respective ports; schedules the ports for the entire packet transfer without interference; and then monitors each block transfer.

#### 3.7.1.1 Data Bus Timing

Data transfers shall be synchronous within specified time slots. The timing cycle is illustrated in Figure 3-6. All ports shall enable automatically during the sync time. The sync time is 4.0  $\mu$ S. This will allow time for the hardwired control lines between the Master Controller and the ports to settle. The data field is 20.48  $\mu$ S long. Within this field, two types of data will exist. Class I data is data flowing from the staging area to the AMM. Class II data is all other DBMS data. Each data type will be transferred at 100 MBS (megabits per second). The two types of data will be time division multiplexed as shown in Figure 3-7. The control field is 1.84  $\mu$ S long which allows 184 bits of information to be transferred. The general format of this field is as described in paragraph 2.5.2.

#### 3.7.1.2 Data Transmission

Data and control words shall be transmitted within the frame synchronization scheme described in paragraph 3.7.1.1. A suitable modulation and synchronization technique shall be employed to be compatible with the concurrent transmission of the two classes of data at 100 megabits per second each. A time divisioned multiplexing scheme at the bit level is illustrated in Figure 3-7. The pulses

## ◀ DATA BUS TIME SLOTS ▶

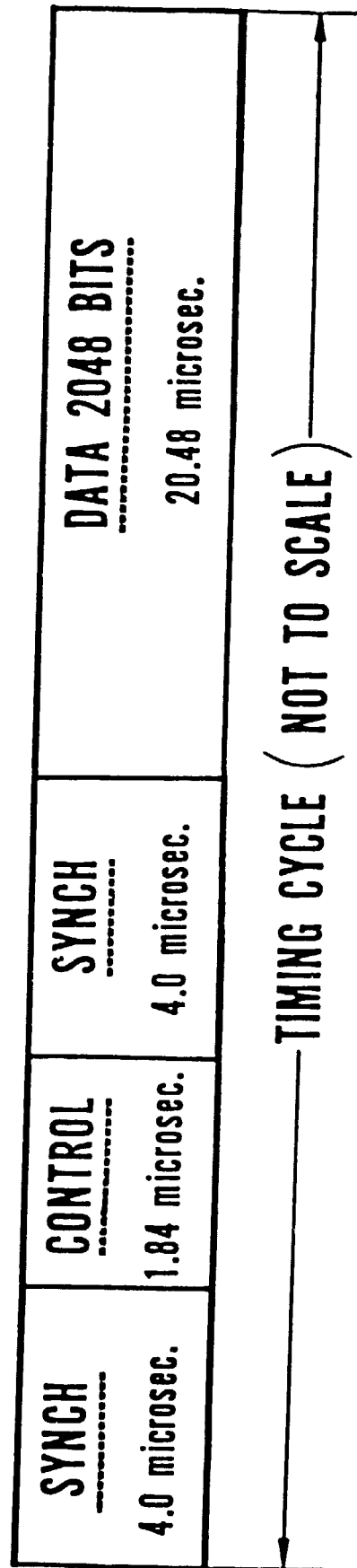


Figure 3-6. Data Bus Timing Cycle

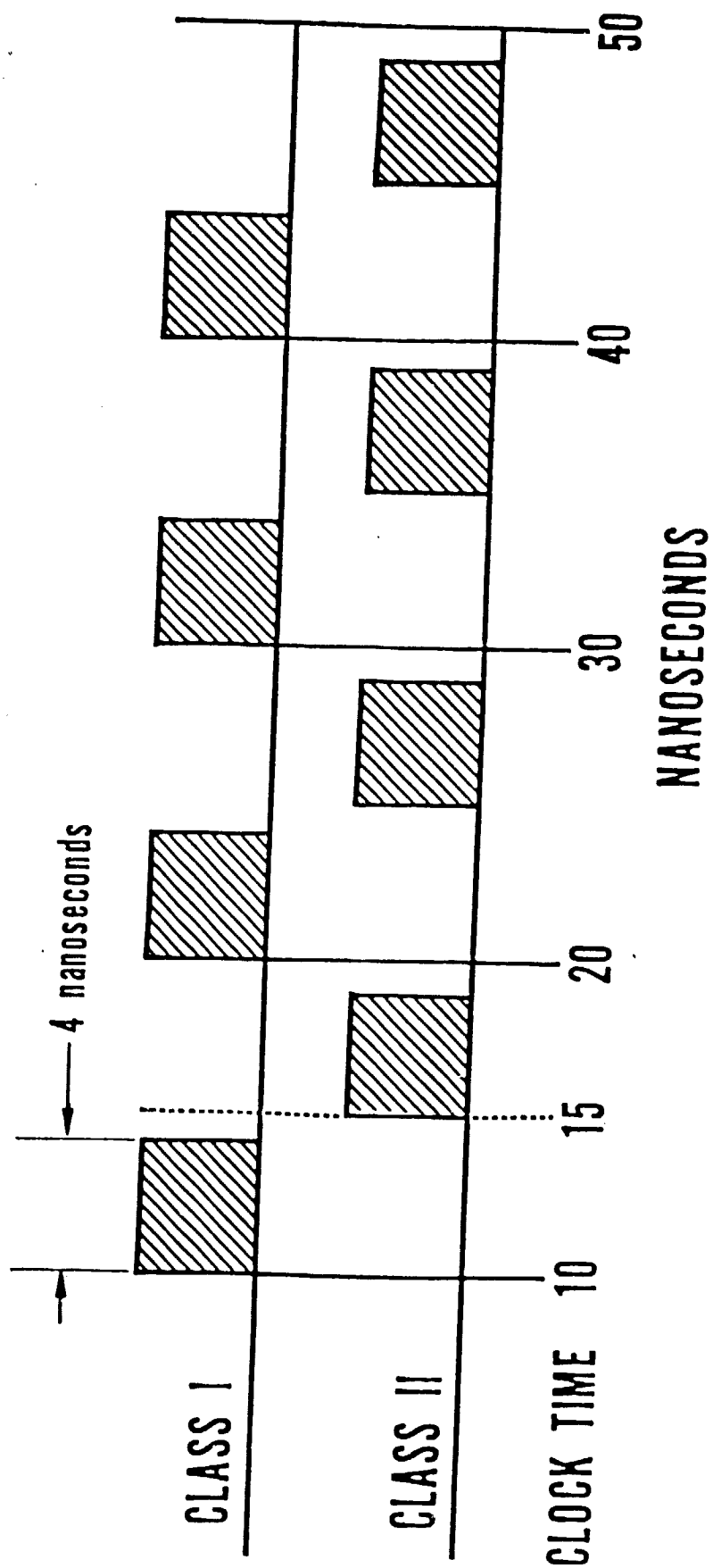


Figure 3-7. Class I and Class II Data

Indicate gating time during which the selected modulation technique is to be employed. On-off-shift keying is illustrated by the pulses of Figure 3-8. A pulse-width modulation, a tone-shift modulation, or techniques that conform to the principle of non-interference with the other class of data transfer shall be acceptable. The choice of modulation technique shall be determined by the synchronization approach selected and the receiver gain requirements dictated by the system loss analysis.

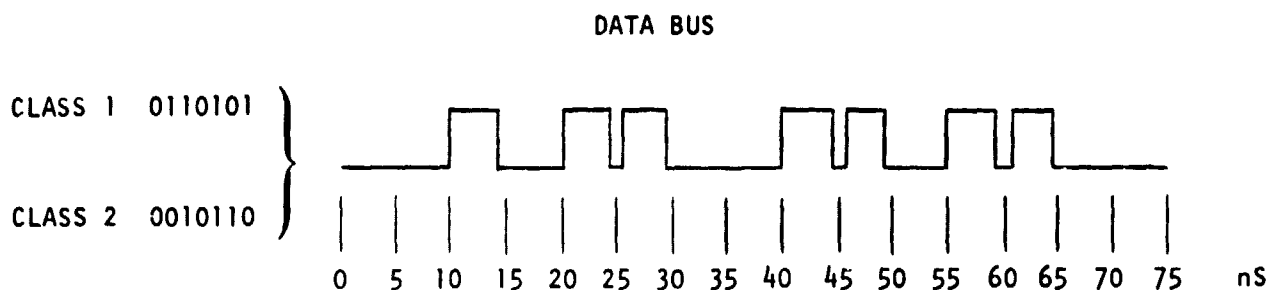


Figure 3-8. Time Division Multiplexed Data Bus

### 3.7.1.3 Bit Error Rate

The overall bit error rate, including all connectors, the transmissive star, and the fiber optic cable shall be no greater than  $10^{-9}$ .

### 3.7.1.4 Data Bus Length

The actual length of each cable between the transmitter or receiver port and the transmissive star shall be determined during detailed system design. The nominal length shall be 50 meters. The system shall be designed to operate according to the transfer and error rates specified herein for a distance of 500 meters between the transmissive star and any port.

## 3.7.2 DATA BUS OPERATION

### 3.7.2.1 Data Transfer

Data entering the port from the data bus shall be as described in paragraph 3.7.1.1. The port shall have the capability to synchronize to it and receive



the 2048 bit burst of data when instructed to. An 8-bit cyclic redundancy check shall be performed on all incoming data. The CRC is for error detection and not error correction. In the event an error is found, the port's acknowledge line shall not be raised signaling the Master Controller that an error was detected. The incoming data will be clocked into a 2048 bit buffer to await transfer to the user. Transfer of data to the user is handled by a programmable interface. During data transfers to or from an interfacing device, the port's inhibit line will be raised signaling the Master Controller the port is not ready to accept or transmit any data.

Data entering the port through the connecting device interface shall be temporarily stored in the 2048 bit buffer. When the buffer is full or an end of a block of data is found, the port will signal the Master Controller it wishes to transmit data. The data, along with an 8-bit CRC word, computed by the Data Bus Port, will be transmitted in the format described in paragraph 3.7.

#### 3.7.2.2 Control Time

The control time is 1.84  $\mu$ S long which will permit 184 bits of control. The ports with their poll lines raised by the Master Controller will accept the control word for decoding; the other ports will ignore the control word. The control word contains instructions for the data bus port to prepare to transmit, prepare to receive or other instructions to process the data. It also contains information defining the connected device user interface.

### 3.8 MASTER CONTROLLER AND DATA BUS PORT

The Master Controller and Data Bus Port provides the interface between VAX 11 and the data bus and controls the data flow on the data bus. This device is illustrated in Figure 3-9.

#### 3.8.1 FUNCTION

The data bus port shall function in a similar manner as described in paragraph 3.7.5. The master controller function shall be as described in paragraph 2.5 through 2.5.3 and 3.7 through 3.7.4.2. Additionally, the master controller shall output a distinct control and data synchronization code during the respective synch times.

# VAX INTERFACE

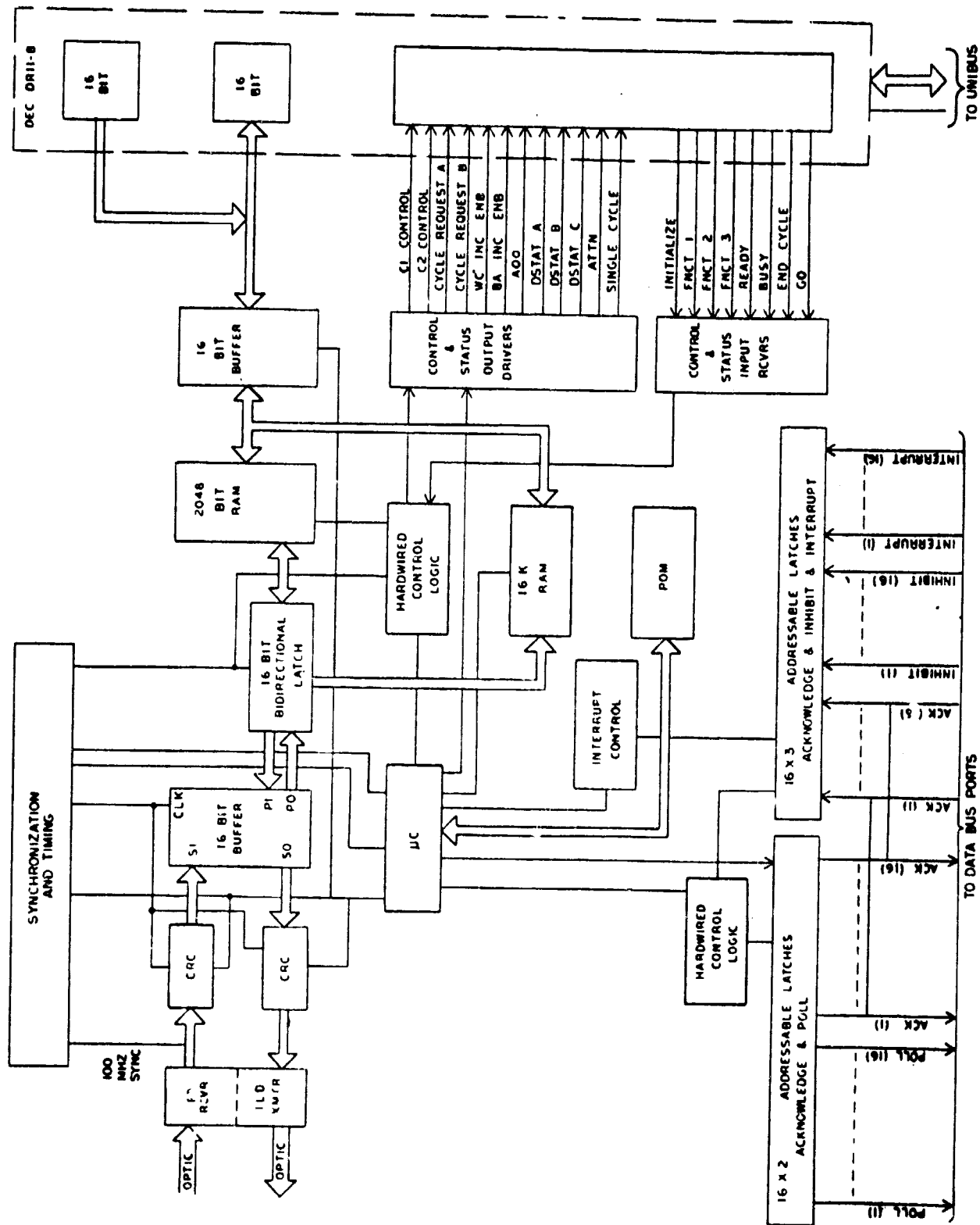


Figure 3-9. VAX Data Bus and Master Controller

### 3.8.2 CONTROL SIGNALS

The Master Controller shall have thirty-two (32) latchable output drivers and forty-eight (48) input receivers (16 latchable). These shall provide independent control and status signals for sixteen (16) data bus ports. The signals shall be TTL compatible capable of driving and receiving signals over 500 meters of twisted pair cable. Adequate noise filtering shall be provided to achieve the system BER identified in Section 5, with a settling time not to exceed the synchronization time of the data bus timing. The output signals shall be control and acknowledge. Input signals shall be acknowledge, inhibit, and interrupt.

#### 3.8.2.1 Poll

This line directs the device port to become active. If the poll line is raised after the MC receives an interrupt, the device port will transmit data; otherwise, it will prepare to receive data. This signal will become active during the data bus sync time.

#### 3.8.2.2 Inhibit

This signal is active whenever the device port is not prepared to accept or transmit data. The signal shall be raised when data is being transferred to or from an interfacing device.

#### 3.8.2.3 Acknowledge

This line will be raised by the device port when a 2048 bit block of data has been received, the CRC decoded, and determined to be correct. If the CRC is incorrect, the acknowledge will not be raised and the Master Controller will take appropriate action.

#### 3.8.2.4 Interrupt

This line signals the Master Controller that the device port has information to transfer. The device port interrupt line shall not become active if its inhibit line is active.

Indicate gating time during which the selected modulation technique is to be employed. On-off-shift keying is illustrated by the pulses of Figure 3-8. A pulse-width modulation, a tone-shift modulation, or techniques that conform to the principle of non-interference with the other class of data transfer shall be acceptable. The choice of modulation technique shall be determined by the synchronization approach selected and the receiver gain requirements dictated by the system loss analysis.

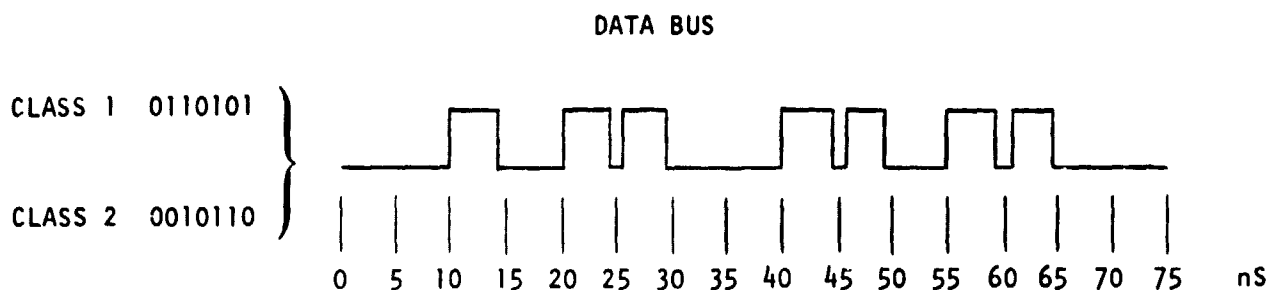


Figure 3-8. Time Division Multiplexed Data Bus

### 3.7.1.3 Bit Error Rate

The overall bit error rate, including all connectors, the transmissive star, and the fiber optic cable shall be no greater than  $10^{-9}$ .

### 3.7.1.4 Data Bus Length

The actual length of each cable between the transmitter or receiver port and the transmissive star shall be determined during detailed system design. The nominal length shall be 50 meters. The system shall be designed to operate according to the transfer and error rates specified herein for a distance of 500 meters between the transmissive star and any port.

## 3.7.2 DATA BUS OPERATION

### 3.7.2.1 Data Transfer

Data entering the port from the data bus shall be as described in paragraph 3.7.1.1. The port shall have the capability to synchronize to it and receive

the 2048 bit burst of data when instructed to. An 8-bit cyclic redundancy check shall be performed on all incoming data. The CRC is for error detection and not error correction. In the event an error is found, the port's acknowledge line shall not be raised signaling the Master Controller that an error was detected. The incoming data will be clocked into a 2048 bit buffer to await transfer to the user. Transfer of data to the user is handled by a programmable interface. During data transfers to or from an interfacing device, the port's inhibit line will be raised signaling the Master Controller the port is not ready to accept or transmit any data.

Data entering the port through the connecting device interface shall be temporarily stored in the 2048 bit buffer. When the buffer is full or an end of a block of data is found, the port will signal the Master Controller it wishes to transmit data. The data, along with an 8-bit CRC word, computed by the Data Bus Port, will be transmitted in the format described in paragraph 3.7.

#### 3.7.2.2 Control Time

The control time is 1.84  $\mu$ s long which will permit 184 bits of control. The ports with their poll lines raised by the Master Controller will accept the control word for decoding; the other ports will ignore the control word. The control word contains instructions for the data bus port to prepare to transmit, prepare to receive or other instructions to process the data. It also contains information defining the connected device user interface.

### 3.8 MASTER CONTROLLER AND DATA BUS PORT

The Master Controller and Data Bus Port provides the interface between VAX 11 and the data bus and controls the data flow on the data bus. This device is illustrated in Figure 3-9.

#### 3.8.1 FUNCTION

The data bus port shall function in a similar manner as described in paragraph 3.7.5. The master controller function shall be as described in paragraph 2.5 through 2.5.3 and 3.7 through 3.7.4.2. Additionally, the master controller shall output a distinct control and data synchronization code during the respective synch times.

# VAX INTERFACE

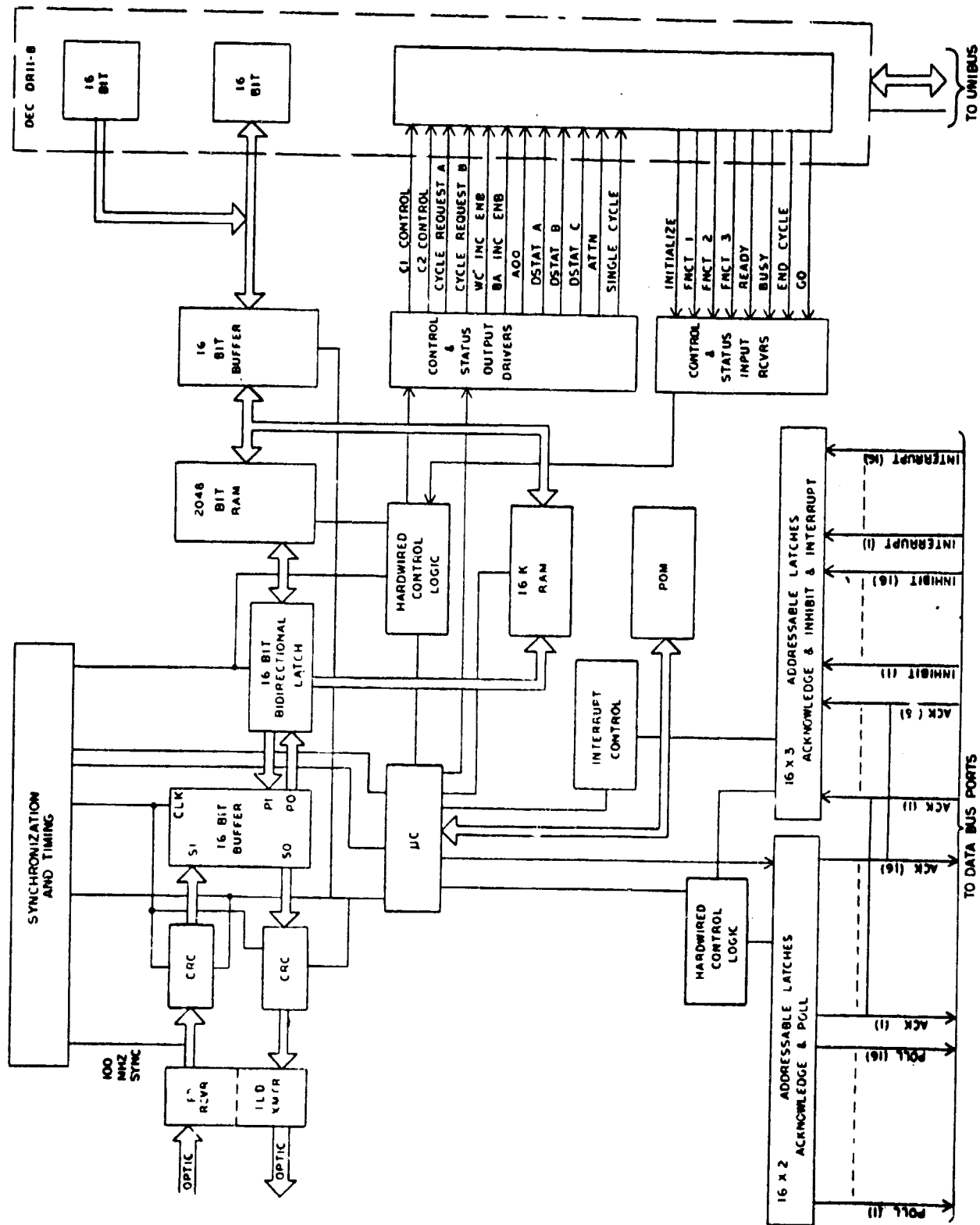


Figure 3-9. VAX Data Bus and Master Controller

### 3.8.2 CONTROL SIGNALS

The Master Controller shall have thirty-two (32) latchable output drivers and forty-eight (48) input receivers (16 latchable). These shall provide independent control and status signals for sixteen (16) data bus ports. The signals shall be TTL compatible capable of driving and receiving signals over 500 meters of twisted pair cable. Adequate noise filtering shall be provided to achieve the system BER identified in Section 5, with a settling time not to exceed the synchronization time of the data bus timing. The output signals shall be control and acknowledge. Input signals shall be acknowledge, inhibit, and interrupt.

#### 3.8.2.1 Poll

This line directs the device port to become active. If the poll line is raised after the MC receives an interrupt, the device port will transmit data; otherwise, it will prepare to receive data. This signal will become active during the data bus sync time.

#### 3.8.2.2 Inhibit

This signal is active whenever the device port is not prepared to accept or transmit data. The signal shall be raised when data is being transferred to or from an interfacing device.

#### 3.8.2.3 Acknowledge

This line will be raised by the device port when a 2048 bit block of data has been received, the CRC decoded, and determined to be correct. If the CRC is incorrect, the acknowledge will not be raised and the Master Controller will take appropriate action.

#### 3.8.2.4 Interrupt

This line signals the Master Controller that the device port has information to transfer. The device port interrupt line shall not become active if its inhibit line is active.

### 3.8.3 SPECIAL HARDWARE

The Master Controller and Data Bus Port shall contain the hardware of a standard data bus port and additional hardware to implement the functions described in paragraph 3.8.2. The following additional hardware is required.

- o 16 K byte Random Access Memory
- o Read Only Memory
- o 32 latchable output drivers
- o 32 input receivers
- o 16 latchable input receivers

The RAM shall be used to accept sequences of command words from the supporting CMS software processors operating in VAX II. It shall also be used to maintain the current status of the bus ports. The bus control sequences shall be implemented in the ROM.

### 3.8.4 MICROCOMPUTER

If necessary, the Master Control function may be implemented in a separate microcomputer from the data bus port microcomputer.

## 3.9 TYPICAL USER DATA BUS PORT

The user data bus port shall serve as the interface between the DBMS fiber optic data bus and a user or group of users. The port receives instructions from the Master Controller, via control words during the control time slot and four hardwire lines. A typical Data Bus Port is illustrated in Figure 3-10. The layout of Figure 3-10 is suggested to reduce the requirements for high speed logical components. Flexibility of where the error detection is performed shall be permitted.

### 3.9.1 FIBER OPTIC INTERFACE

The User Data Bus Port shall communicate with other units within the DBMS via the Fiber Optic Data Bus. The User Data Bus Port shall transmit and receive only Class II data as described in paragraph 2.5.1. The fiber optic receiver and transmitter shall receive and transmit data at a rate of 100 MBS



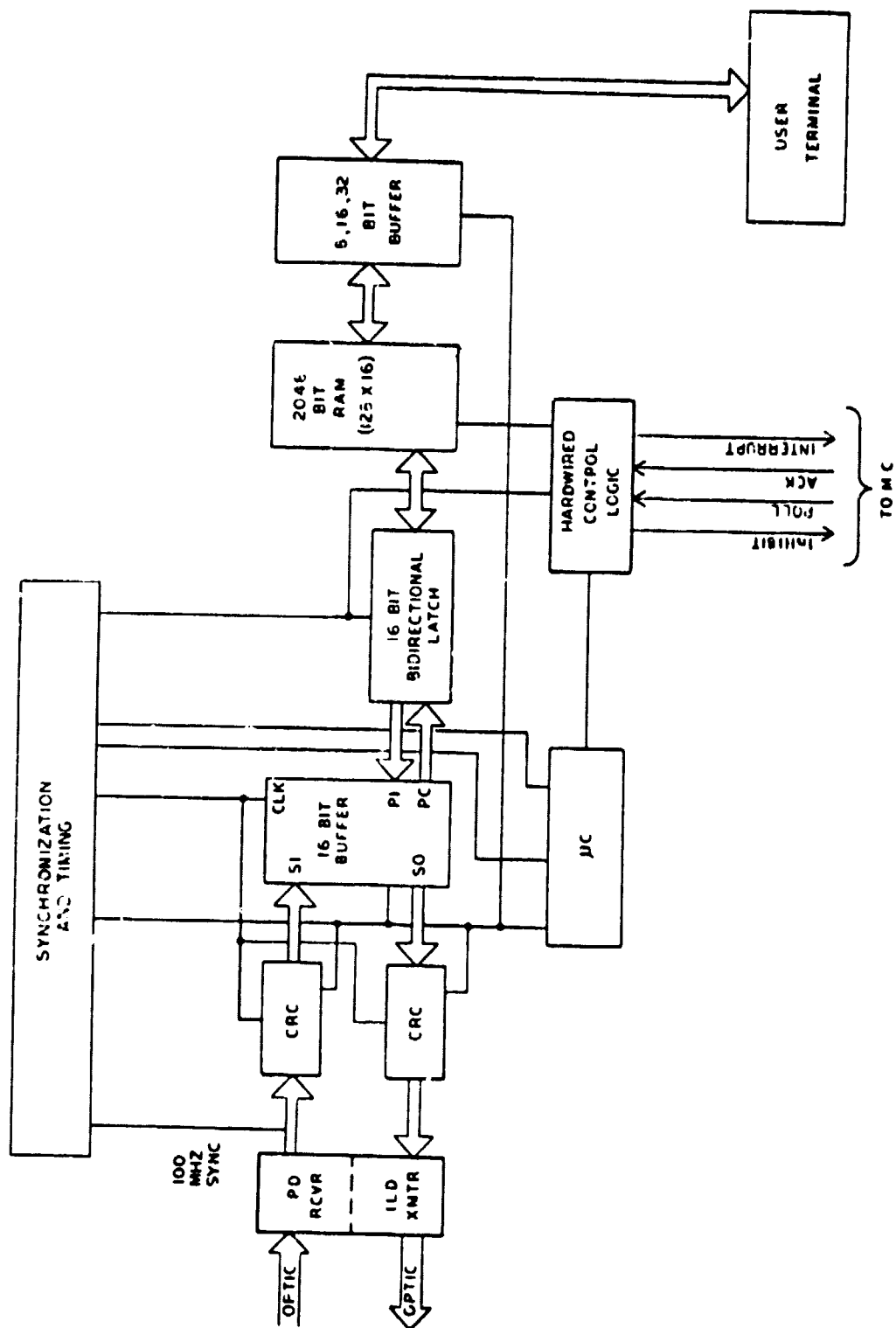


Figure 3-10. User Terminal Data Bus Port

but shall operate in a 250 MHz bandwidth. This is described in paragraph 3.7.1.1. All data transfers shall take place in blocks of a maximum of 2048 bits.

### 3.9.2 DEVICE INTERFACE

The interface to the user shall be programmable to allow several different users to be connected to the system. It shall allow bit serial or up to 32 bit parallel data transfers. The user data bus port shall use a microprocessor or other computing device to provide the programmable interface.

### 3.9.3 MASTER CONTROLLER INTERFACE

The device data bus port shall have four lines connecting it to the Master Controller: poll, inhibit, acknowledge, and interrupt. The function of each signal is described in paragraphs 3.8.2.1 through 3.8.2.4.

### 3.10 VAX DATA BUS PORT

The fiber optic data bus port shall interface to a DEC DR-11B which in turn interfaces to the VAX UNIBUS. The operation of the port is the same as described in paragraphs 3.7.2 through 3.7.4.2 except the device interface has been defined as the DR11-B interface. The interface to both VAX I and VAX II is the same. The interface to VAX I is illustrated in Figure 3-11.

### 3.11 AUXILIARY STORAGE DATA BUS PORT

The auxiliary storage data bus port shall serve as the interface between the fiber optic data bus and the auxiliary storage high speed interface. The operation of the port shall be the same as a user data bus port except the user interface has been defined as the high speed parallel data interface of the AMPEX DCP-909 Parallel Transfer Drive Control Unit. Details of this interface may be obtained from AMPEX Engineering Specification 3309527-01. This port is illustrated in Figure 3-12.

### 3.12 ARCHIVAL MASS MEMORY DATA BUS PORT

The archival mass memory data bus port shall serve as the interface between the fiber optic data bus and the archival mass memory. Because the AMM can both

Indicate gating time during which the selected modulation technique is to be employed. On-off-shift keying is illustrated by the pulses of Figure 3-8. A pulse-width modulation, a tone-shift modulation, or techniques that conform to the principle of non-interference with the other class of data transfer shall be acceptable. The choice of modulation technique shall be determined by the synchronization approach selected and the receiver gain requirements dictated by the system loss analysis.

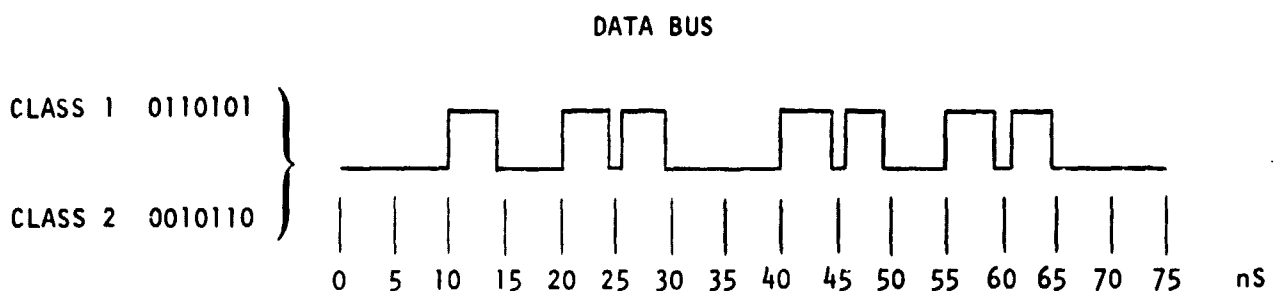


Figure 3-8. Time Division Multiplexed Data Bus

### 3.7.1.3 Bit Error Rate

The overall bit error rate, including all connectors, the transmissive star, and the fiber optic cable shall be no greater than  $10^{-9}$ .

### 3.7.1.4 Data Bus Length

The actual length of each cable between the transmitter or receiver port and the transmissive star shall be determined during detailed system design. The nominal length shall be 50 meters. The system shall be designed to operate according to the transfer and error rates specified herein for a distance of 500 meters between the transmissive star and any port.

## 3.7.2 DATA BUS OPERATION

### 3.7.2.1 Data Transfer

Data entering the port from the data bus shall be as described in paragraph 3.7.1.1. The port shall have the capability to synchronize to it and receive

the 2048 bit burst of data when instructed to. An 8-bit cyclic redundancy check shall be performed on all incoming data. The CRC is for error detection and not error correction. In the event an error is found, the port's acknowledge line shall not be raised signaling the Master Controller that an error was detected. The incoming data will be clocked into a 2048 bit buffer to await transfer to the user. Transfer of data to the user is handled by a programmable interface. During data transfers to or from an interfacing device, the port's inhibit line will be raised signaling the Master Controller the port is not ready to accept or transmit any data.

Data entering the port through the connecting device interface shall be temporarily stored in the 2048 bit buffer. When the buffer is full or an end of a block of data is found, the port will signal the Master Controller it wishes to transmit data. The data, along with an 8-bit CRC word, computed by the Data Bus Port, will be transmitted in the format described in paragraph 3.7.

#### 3.7.2.2 Control Time

The control time is 1.84  $\mu$ S long which will permit 184 bits of control. The ports with their poll lines raised by the Master Controller will accept the control word for decoding; the other ports will ignore the control word. The control word contains instructions for the data bus port to prepare to transmit, prepare to receive or other instructions to process the data. It also contains information defining the connected device user interface.

### 3.8 MASTER CONTROLLER AND DATA BUS PORT

The Master Controller and Data Bus Port provides the interface between VAX 11 and the data bus and controls the data flow on the data bus. This device is illustrated in Figure 3-9.

#### 3.8.1 FUNCTION

The data bus port shall function in a similar manner as described in paragraph 3.7.5. The master controller function shall be as described in paragraph 2.5 through 2.5.3 and 3.7 through 3.7.4.2. Additionally, the master controller shall output a distinct control and data synchronization code during the respective synch times.

# VAX INTERFACE

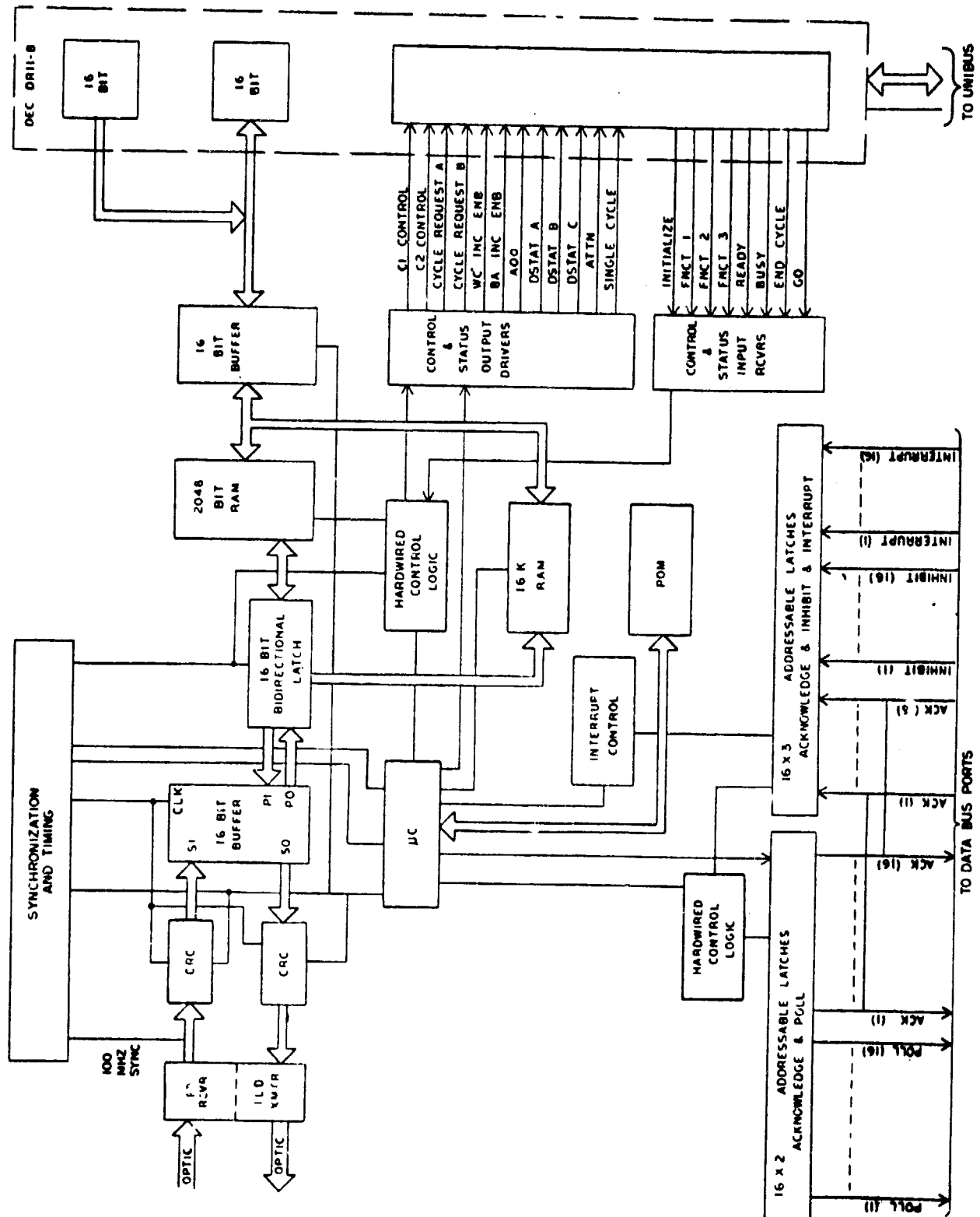


Figure 3-9. VAX Data Bus and Master Controller

### 3.8.2 CONTROL SIGNALS

The Master Controller shall have thirty-two (32) latchable output drivers and forty-eight (48) input receivers (16 latchable). These shall provide independent control and status signals for sixteen (16) data bus ports. The signals shall be TTL compatible capable of driving and receiving signals over 500 meters of twisted pair cable. Adequate noise filtering shall be provided to achieve the system BER identified in Section 5, with a settling time not to exceed the synchronization time of the data bus timing. The output signals shall be control and acknowledge. Input signals shall be acknowledge, inhibit, and interrupt.

#### 3.8.2.1 Poll

This line directs the device port to become active. If the poll line is raised after the MC receives an interrupt, the device port will transmit data; otherwise, it will prepare to receive data. This signal will become active during the data bus sync time.

#### 3.8.2.2 Inhibit

This signal is active whenever the device port is not prepared to accept or transmit data. The signal shall be raised when data is being transferred to or from an interfacing device.

#### 3.8.2.3 Acknowledge

This line will be raised by the device port when a 2048 bit block of data has been received, the CRC decoded, and determined to be correct. If the CRC is incorrect, the acknowledge will not be raised and the Master Controller will take appropriate action.

#### 3.8.2.4 Interrupt

This line signals the Master Controller that the device port has information to transfer. The device port interrupt line shall not become active if its inhibit line is active.

### 3.8.3 SPECIAL HARDWARE

The Master Controller and Data Bus Port shall contain the hardware of a standard data bus port and additional hardware to implement the functions described in paragraph 3.8.2. The following additional hardware is required.

- o 16 K byte Random Access Memory
- o Read Only Memory
- o 32 latchable output drivers
- o 32 input receivers
- o 16 latchable input receivers

The RAM shall be used to accept sequences of command words from the supporting CMS software processors operating in VAX II. It shall also be used to maintain the current status of the bus ports. The bus control sequences shall be implemented in the ROM.

### 3.8.4 MICROCOMPUTER

If necessary, the Master Control function may be implemented in a separate microcomputer from the data bus port microcomputer.

## 3.9 TYPICAL USER DATA BUS PORT

The user data bus port shall serve as the interface between the DBMS fiber optic data bus and a user or group of users. The port receives instructions from the Master Controller, via control words during the control time slot and four hardwire lines. A typical Data Bus Port is illustrated in Figure 3-10. The layout of Figure 3-10 is suggested to reduce the requirements for high speed logical components. Flexibility of where the error detection is performed shall be permitted.

### 3.9.1 FIBER OPTIC INTERFACE

The User Data Bus Port shall communicate with other units within the DBMS via the Fiber Optic Data Bus. The User Data Bus Port shall transmit and receive only Class II data as described in paragraph 2.5.1. The fiber optic receiver and transmitter shall receive and transmit data at a rate of 100 MBS

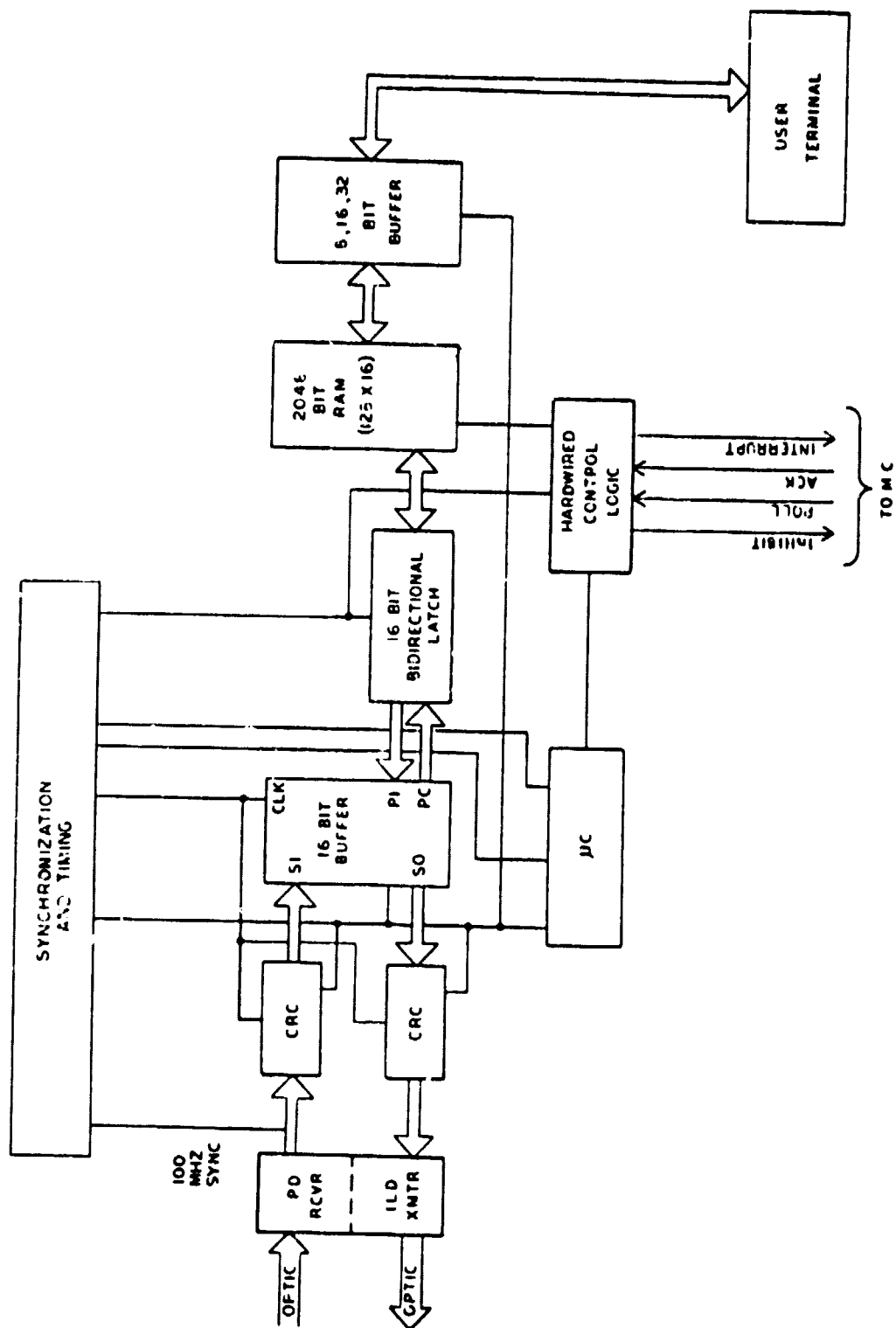


Figure 3-10. User Terminal Data Bus Port



but shall operate in a 250 MHz bandwidth. This is described in paragraph 3.7.1.1. All data transfers shall take place in blocks of a maximum of 2048 bits.

### 3.9.2 DEVICE INTERFACE

The interface to the user shall be programmable to allow several different users to be connected to the system. It shall allow bit serial or up to 32 bit parallel data transfers. The user data bus port shall use a microprocessor or other computing device to provide the programmable interface.

### 3.9.3 MASTER CONTROLLER INTERFACE

The device data bus port shall have four lines connecting it to the Master Controller: poll, inhibit, acknowledge, and interrupt. The function of each signal is described in paragraphs 3.8.2.1 through 3.8.2.4.

### 3.10 VAX DATA BUS PORT

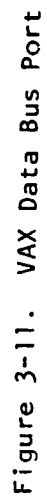
The fiber optic data bus port shall interface to a DEC DR-11B which in turn interfaces to the VAX UNIBUS. The operation of the port is the same as described in paragraphs 3.7.2 through 3.7.4.2 except the device interface has been defined as the DR11-B interface. The interface to both VAX I and VAX II is the same. The interface to VAX I is illustrated in Figure 3-11.

### 3.11 AUXILIARY STORAGE DATA BUS PORT

The auxiliary storage data bus port shall serve as the interface between the fiber optic data bus and the auxiliary storage high speed interface. The operation of the port shall be the same as a user data bus port except the user interface has been defined as the high speed parallel data interface of the AMPEX DCP-909 Parallel Transfer Drive Control Unit. Details of this interface may be obtained from AMPEX Engineering Specification 3309527-01. This port is illustrated in Figure 3-12.

### 3.12 ARCHIVAL MASS MEMORY DATA BUS PORT

The archival mass memory data bus port shall serve as the interface between the fiber optic data bus and the archival mass memory. Because the AMM can both



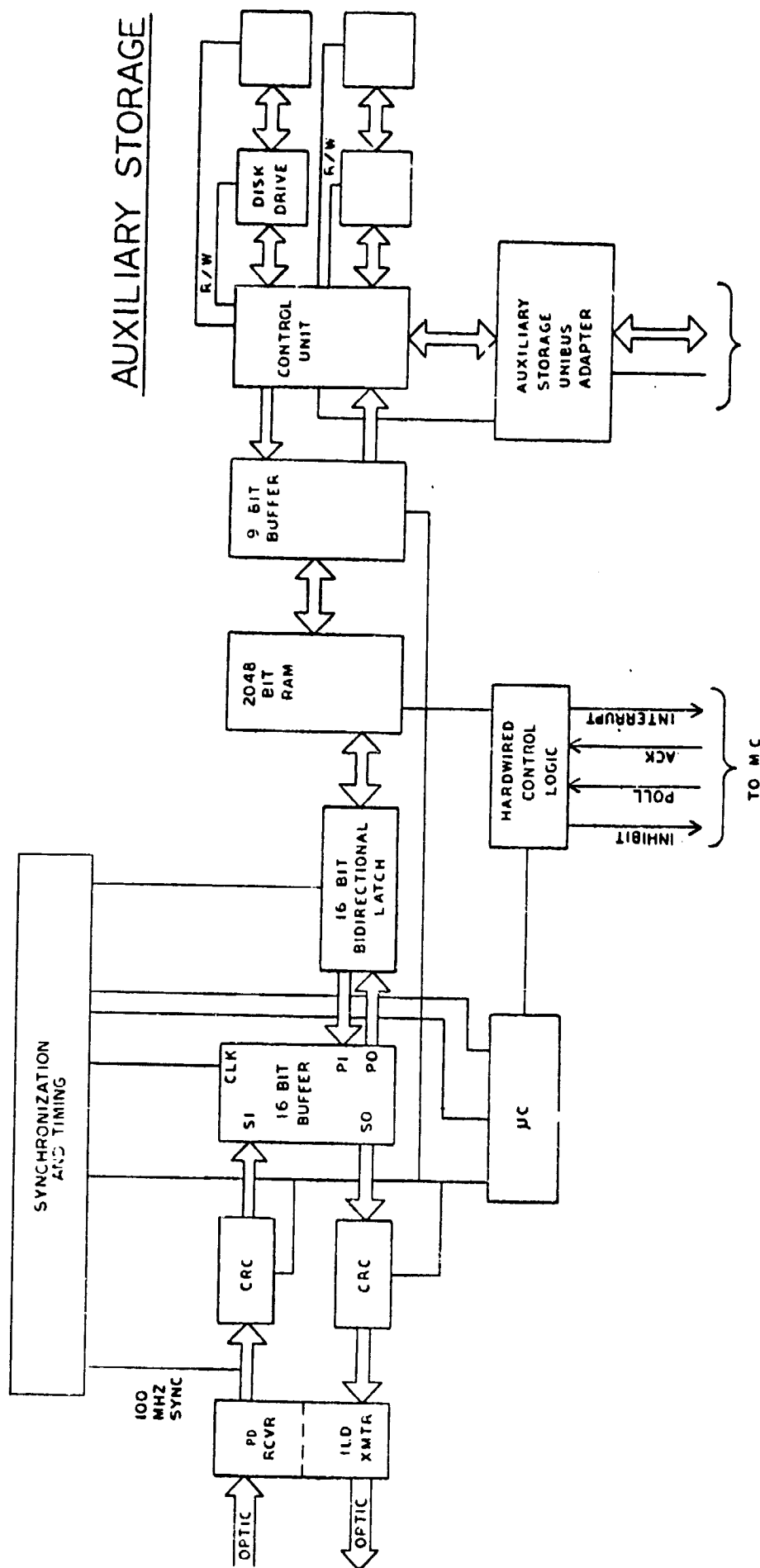


Figure 3-12. Auxiliary Storage Data Bus Port

receive and output data at the same time, the port shall have two 2048 bit buffers to accommodate this capability. Data transmissions to and from the AMM will be in 32-bit wide words. Control and status lines of the interface will be the same as the DR-11B UNIBUS interface. This is illustrated in Figure 3-13. The transfer rate shall be a minimum of 50 megabits per second.

### 3.13 STAGING AREA INTERFACE DATA BUS PORT

The staging area interface port is actually two ports. One port is dedicated to the transmission of Class I data from the staging area to the AMM. The other transmits header data and is responsive to DBMS function commands.

#### 3.13.1 INSTRUMENT PACKET PORT

This port differs from the typical DB port in that it has eight 2048-bit buffers. These eight buffers are filled up with data from the staging area before it is transferred to the AMM. This accommodates the error checking of the X.25 protocol in the staging area. This is illustrated in Figure 3-14.

#### 3.13.2 INSTRUMENT PACKET HEADER PORT

This port operates the same as the user port except that its interface is dedicated to the staging area interface and is not programmable.

### 3.14 MULTIPLEXED DATA BUS PORT

The multiplexed data bus port interfaces the fiber optic data bus to up to eight user devices without a separate multiplexer. The devices may be user computers, terminals, peripheral equipment, or remote communication terminals with their independent protocol. The interfacing formats, serial or parallel may also be different for each port. Where a single user port contains one 2048 bit buffer, the multiplexed data bus port contains eight. Each 2048 bit buffer has associated with it a dedicated microcomputer for interfacing to each particular user. A central microcomputer port controller will control transfers to and from the fiber optic data bus. A ninth 2048 bit buffer will be used to store the control words without tampering with the eight user buffers. On data transfers within the DBMS to a user, a previously received control word will have specified which user is to receive data. While data

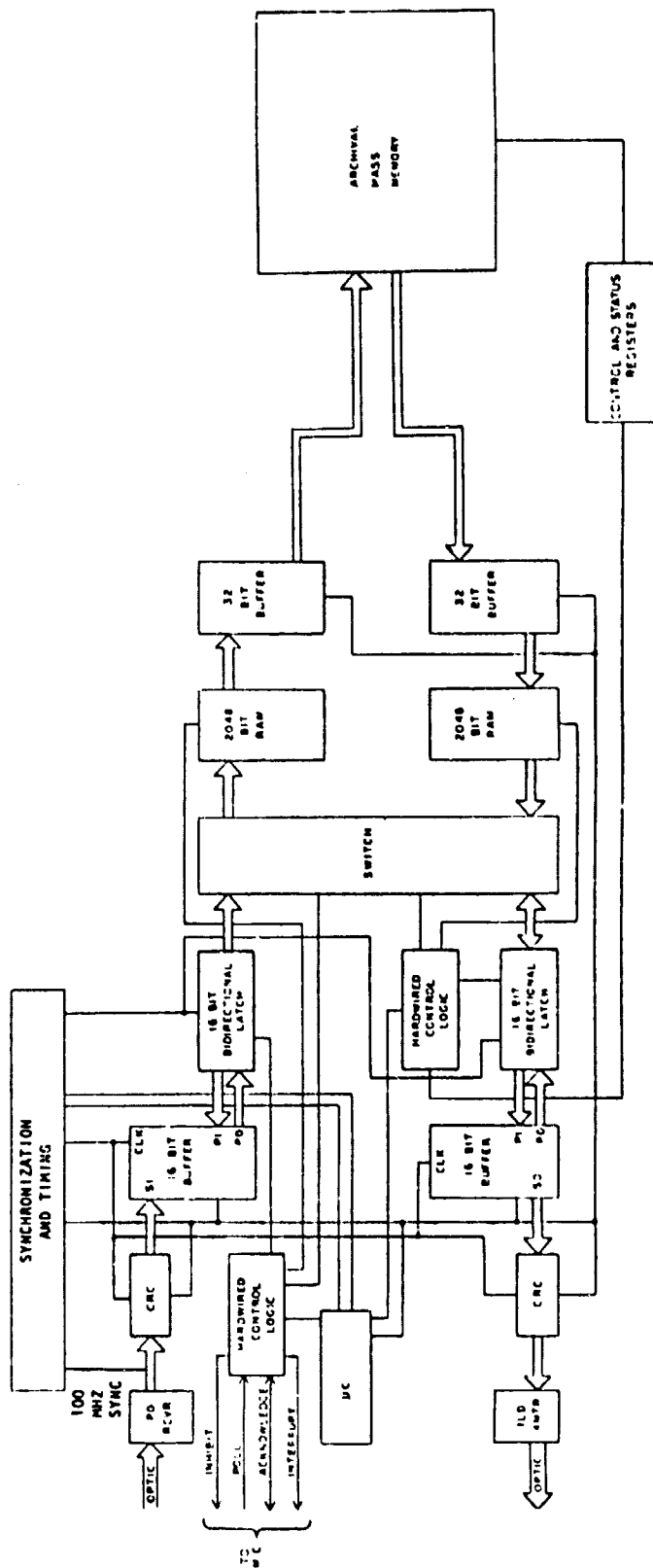


Figure 3-13. Archival Mass Memory Data Bus Port

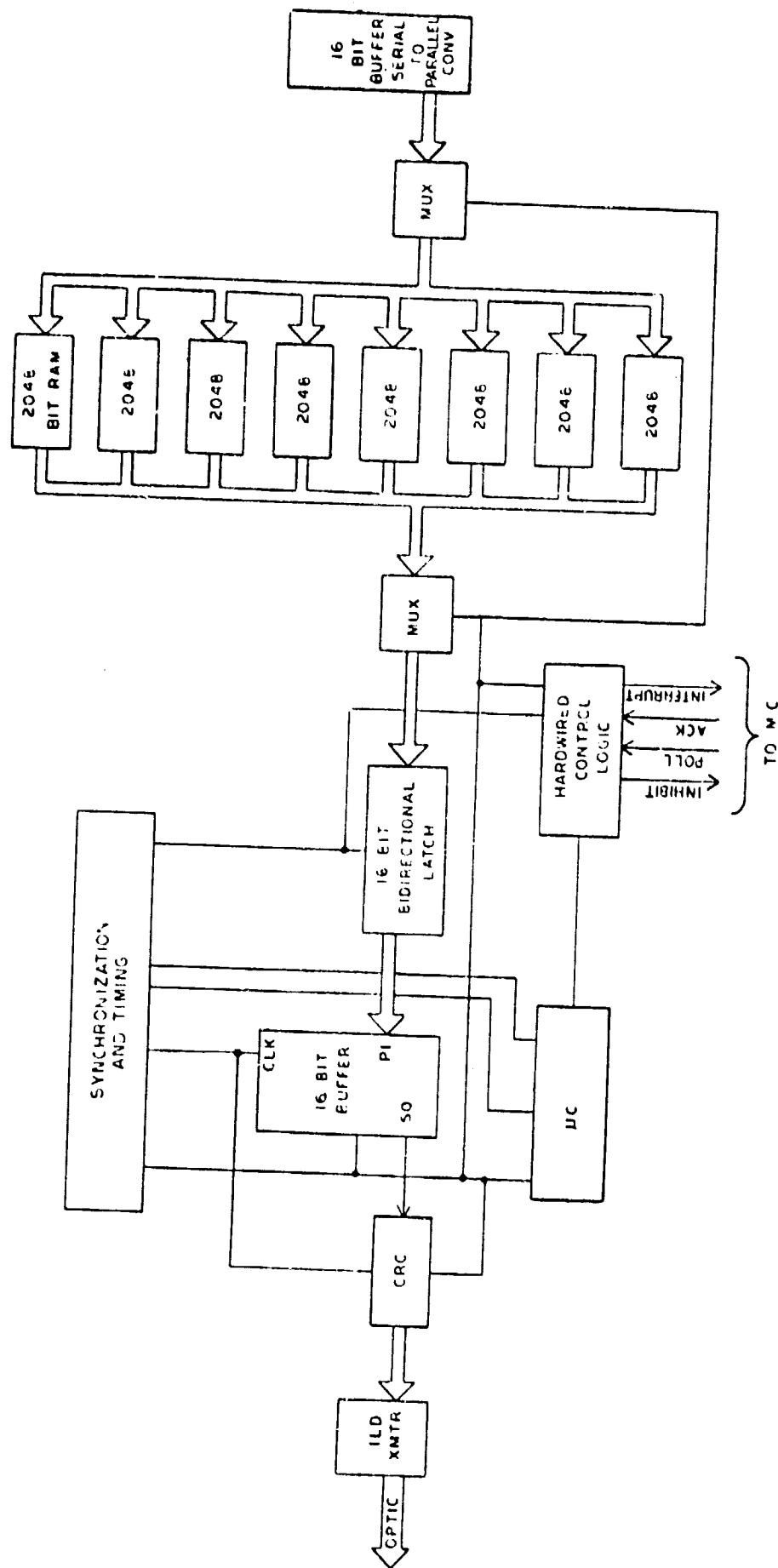


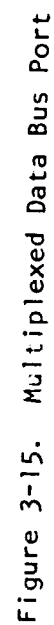
Figure 3-14. Instrument Packet Data Bus Port

is being transferred to a user from its associated buffer, data can be transmitted or received from any other user connected to the port. The dedicated microcomputer for each user will signal the port controller when it is busy. In the event that a particular user is accessed while data is being transferred, the associated microcomputer will signal the port controller that it is busy. The port controller will in turn signal the master controller that it is busy by not lowering its inhibit line. This data bus port is illustrated in Figure 3-15.

The multiplexed data bus port shall be implemented in a modular manner. The interfacing microprocessor, the 2K buffer, and the control and status register for each port shall be field installable. The initial device shall contain the data bus interface and two (2) plug replaceable channels.

### 3.15 MULTIPLEXER

The multiplexer provides an interface which permits multiple devices (up to 8) to share a typical data bus port. The multiplexer typically will accommodate A/N and vector data terminals, or other devices with standardized RS232 interfaces. The multiplexer will serve as the interface between the group of users and the typical data bus port. The multiplexer shall have priority resolution logic for use in the transmission of data from a user to DBMS. The multiplexer shall have address decoding capability for determining which user is to receive data from the DBMS. Addressing shall be performed by the data bus port microprocessor. The multiplexing shall be transparent to the data formatting functions in the DBMS. The microcomputer shall receive the instructions for implementing the addressing as command and data over the data bus.



**Figure 3-15. Multiplexed Data Bus Port**



# VAX INTERFACE

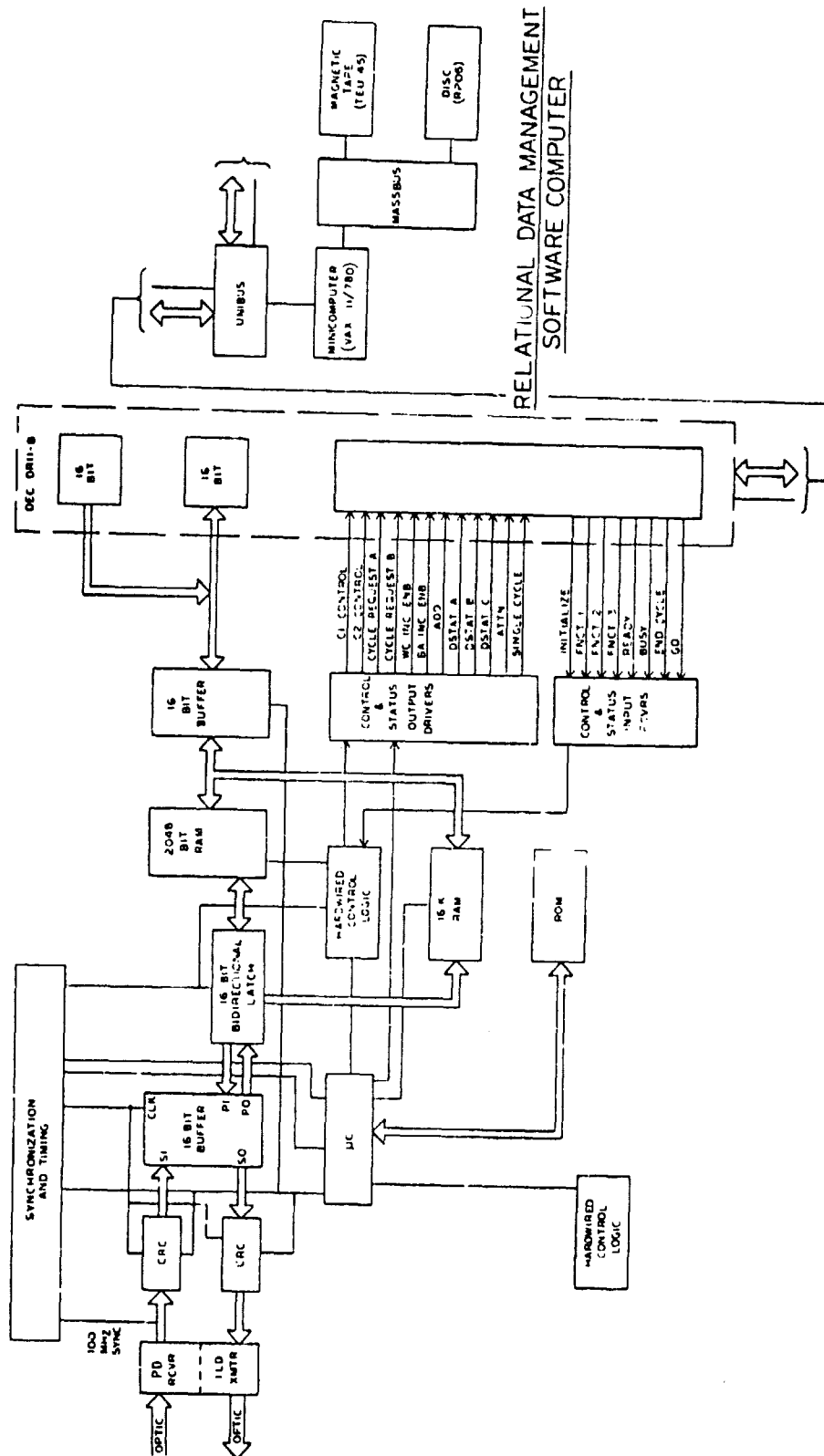


Figure 3-11. VAX Data Bus Port

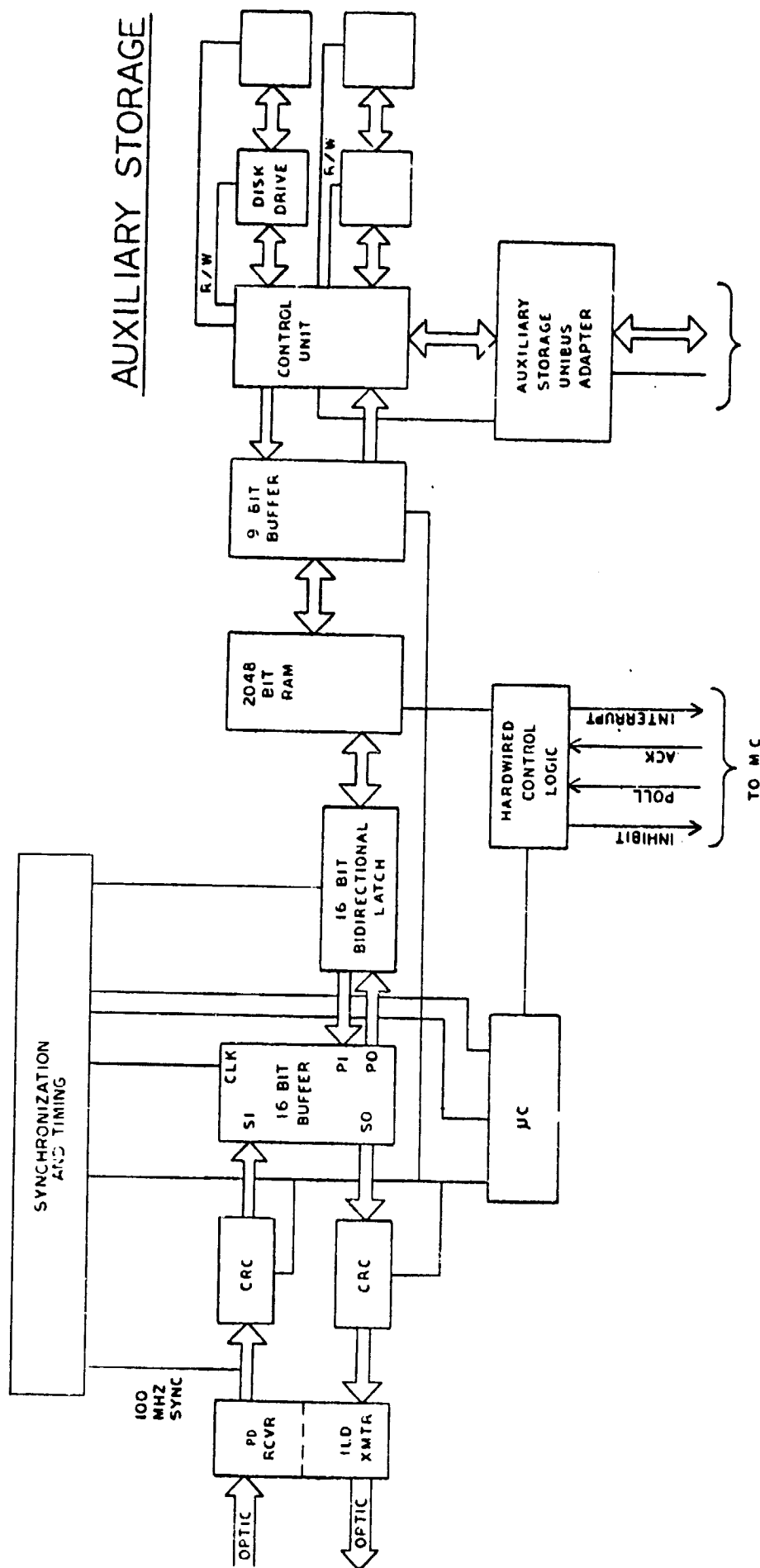


Figure 3-12. Auxiliary Storage Data Bus Port

receive and output data at the same time, the port shall have two 2048 bit buffers to accommodate this capability. Data transmissions to and from the AMM will be in 32-bit wide words. Control and status lines of the interface will be the same as the DR-11B UNIBUS interface. This is illustrated in Figure 3-13. The transfer rate shall be a minimum of 50 megabits per second.

### 3.13 STAGING AREA INTERFACE DATA BUS PORT

The staging area interface port is actually two ports. One port is dedicated to the transmission of Class I data from the staging area to the AMM. The other transmits header data and is responsive to DBMS function commands.

#### 3.13.1 INSTRUMENT PACKET PORT

This port differs from the typical DB port in that it has eight 2048-bit buffers. These eight buffers are filled up with data from the staging area before it is transferred to the AMM. This accommodates the error checking of the X.25 protocol in the staging area. This is illustrated in Figure 3-14.

#### 3.13.2 INSTRUMENT PACKET HEADER PORT

This port operates the same as the user port except that its interface is dedicated to the staging area interface and is not programmable.

### 3.14 MULTIPLEXED DATA BUS PORT

The multiplexed data bus port interfaces the fiber optic data bus to up to eight user devices without a separate multiplexer. The devices may be user computers, terminals, peripheral equipment, or remote communication terminals with their independent protocol. The interfacing formats, serial or parallel may also be different for each port. Where a single user port contains one 2048 bit buffer, the multiplexed data bus port contains eight. Each 2048 bit buffer has associated with it a dedicated microcomputer for interfacing to each particular user. A central microcomputer port controller will control transfers to and from the fiber optic data bus. A ninth 2048 bit buffer will be used to store the control words without tampering with the eight user buffers. On data transfers within the DBMS to a user, a previously received control word will have specified which user is to receive data. While data

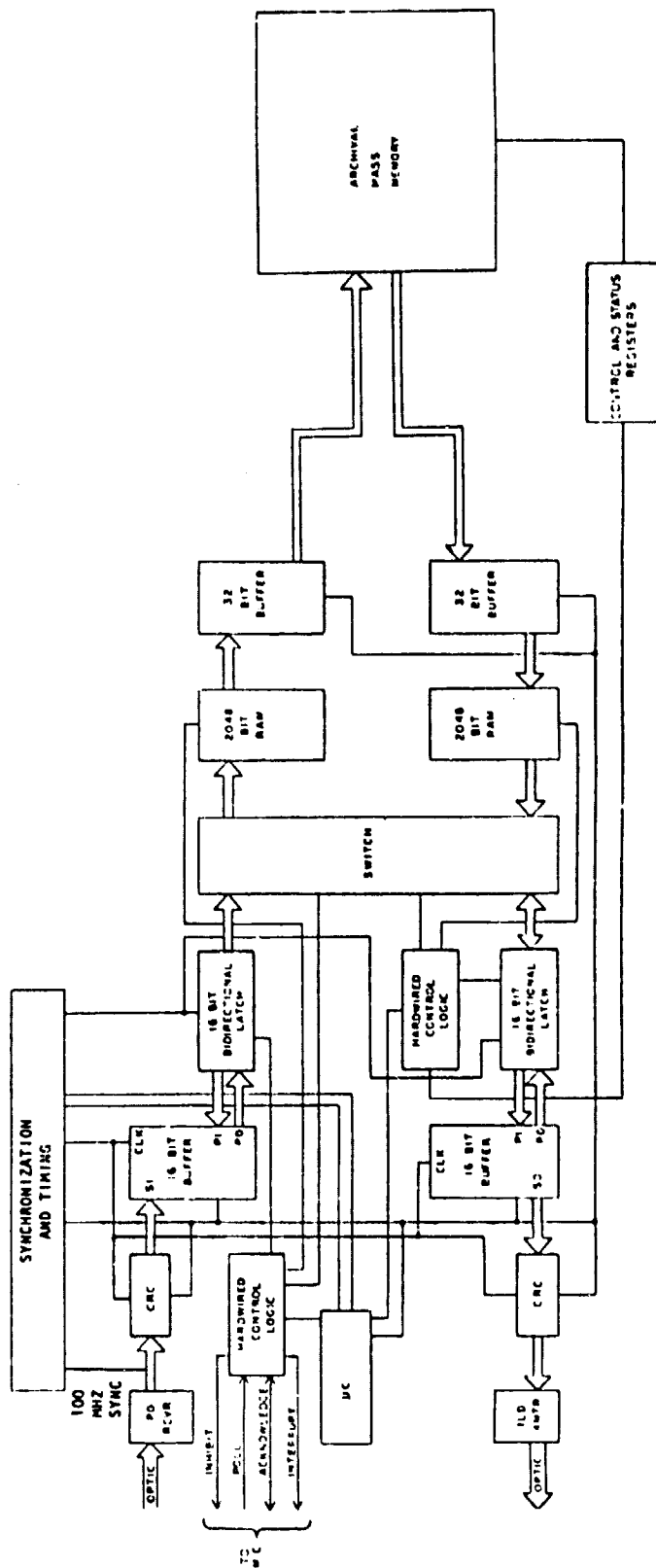


Figure 3-13. Archival Mass Memory Data Bus Port

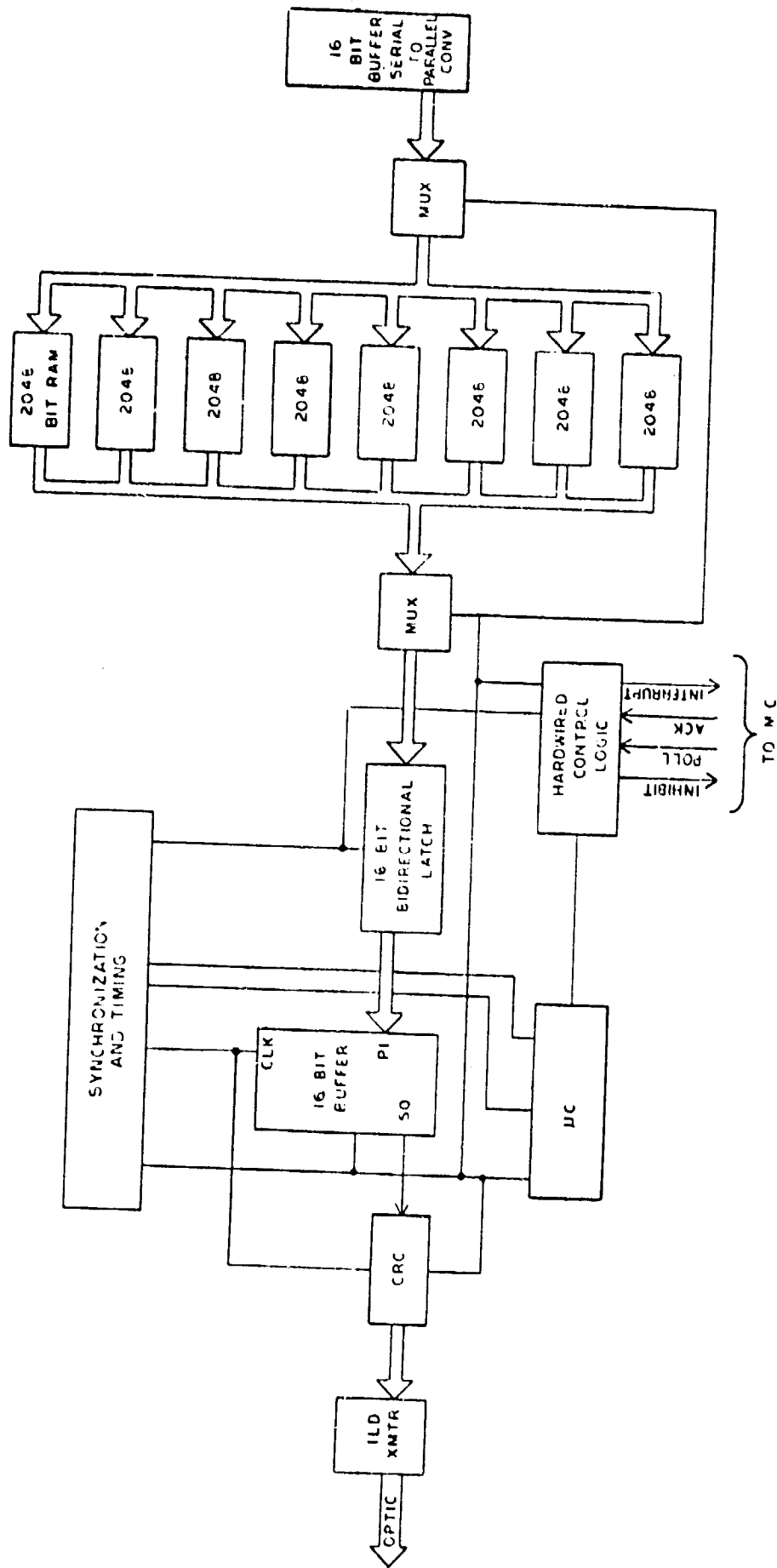


Figure 3-14. Instrument Packet Data Bus Port

is being transferred to a user from its associated buffer, data can be transmitted or received from any other user connected to the port. The dedicated microcomputer for each user will signal the port controller when it is busy. In the event that a particular user is accessed while data is being transferred, the associated microcomputer will signal the port controller that it is busy. The port controller will in turn signal the master controller that it is busy by not lowering its inhibit line. This data bus port is illustrated in Figure 3-15.

The multiplexed data bus port shall be implemented in a modular manner. The interfacing microprocessor, the 2K buffer, and the control and status register for each port shall be field installable. The initial device shall contain the data bus interface and two (2) plug replaceable channels.

### 3.15 MULTIPLEXER

The multiplexer provides an interface which permits multiple devices (up to 8) to share a typical data bus port. The multiplexer typically will accommodate A/N and vector data terminals, or other devices with standardized RS232 interfaces. The multiplexer will serve as the interface between the group of users and the typical data bus port. The multiplexer shall have priority resolution logic for use in the transmission of data from a user to DBMS. The multiplexer shall have address decoding capability for determining which user is to receive data from the DBMS. Addressing shall be performed by the data bus port microprocessor. The multiplexing shall be transparent to the data formatting functions in the DBMS. The microcomputer shall receive the instructions for implementing the addressing as command and data over the data bus.

# PROGRAMMABLE INTERFACES

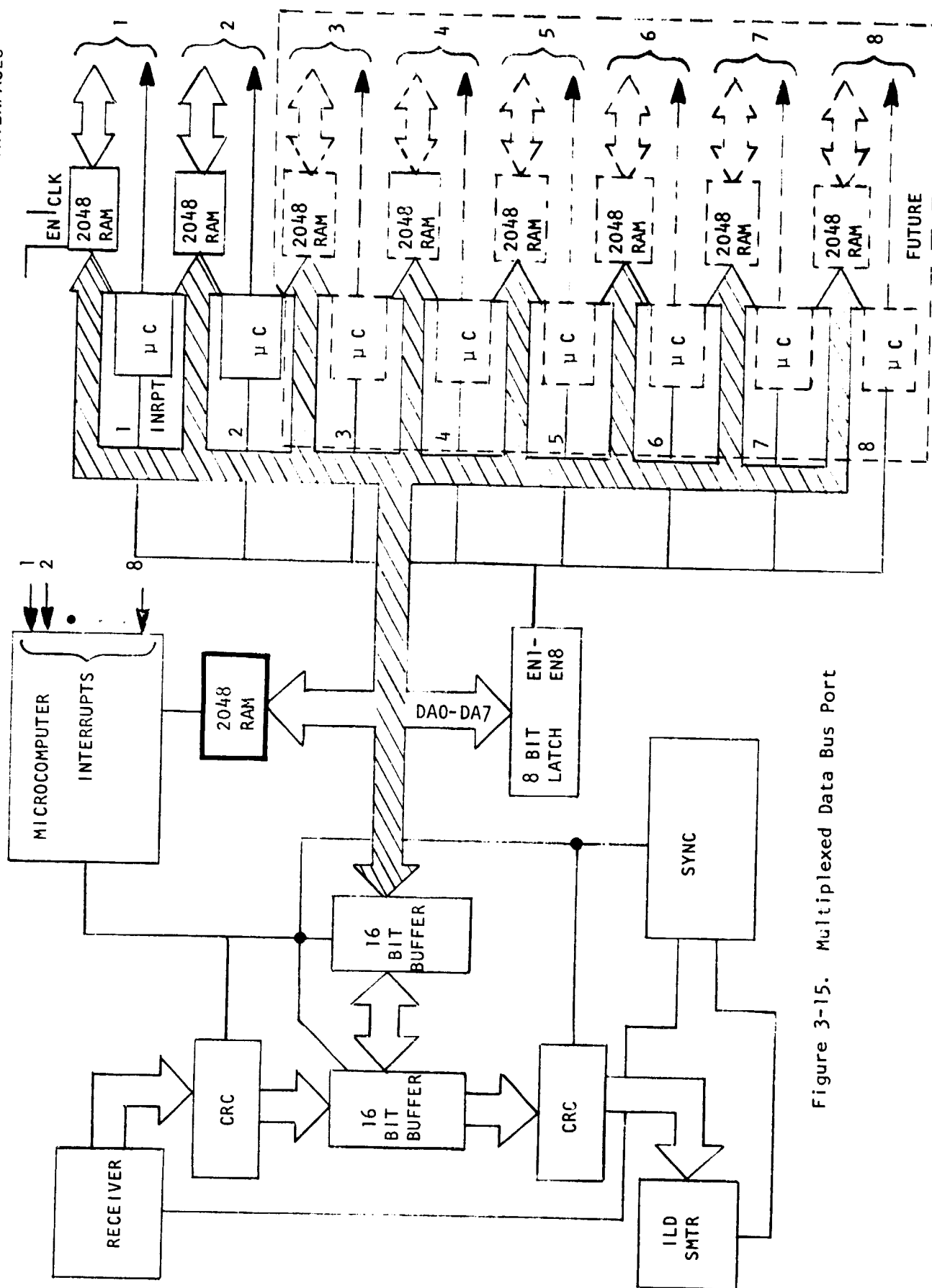


Figure 3-15. Multiplexed Data Bus Port

## SECTION 4.0

### SOFTWARE SPECIFICATION

#### 4.1 CONFIGURATION DATA MANAGEMENT SOFTWARE

The Configuration Management Software (CMS) shall consist of software residing in VAX I, VAX II, and microcode for the microprocessors to control the special purpose hardware devices described in section 3.0.

#### 4.2 SOFTWARE DEFINITION AND GUIDELINES

The following software terminology is included in the Data Base Management System specification. The purpose is to develop a common language defining the following software categories: Operating System, Support Processors, Diagnostics, Special System Software, and Applications Processors. The DBMS software is organized into these five categories that are defined in paragraphs 4.2.1 through 4.2.5.

##### 4.2.1 OPERATING SYSTEM SOFTWARE

This software is normally supplied by the computer manufacturer. In this instance, it is the VAX/VMS. It also includes additional drivers for DBMS unique devices. These are the data bus port, the master controller and data bus port, display controller, and auxiliary storage controller. The operating systems for VAX I and VAX II are identified in paragraphs 4.3 and 4.4, respectively.

##### 4.2.2 SUPPORT PROCESSORS

Support Processors comprise those tools that aid the development, checking, and maintenance of the software system and application processes. It is generally supplied by the computer manufacturers, but is not limited to only that software. Specific processes included are identified in paragraph 4.5. "Processor" is used synonymously with the word "Process." Processors include the assemblers, compilers, editors, linkers, library utilities, debug aids and special function packages. They also include the general purpose display processing that is application independent but necessary to facilitate the display of image data.



### 4.2.3 DIAGNOSTICS

Diagnostic processes provide for the regular exercise and monitoring of system components to determine the health of the system and to aid in the location of defective components. They include but are not limited to exercisers supplied by the computer manufacturer for the standard peripheral equipment such as discs, line printer, memory and CRT terminals. Diagnostics also include processors to verify the proper performance of a device as well as to aid in the isolation of faults. They include software supplied in conjunction with devices such as image displays and specially written simulators to verify interfaces to external systems such as to the staging area.

### 4.2.4 SPECIAL SYSTEM SOFTWARE

This software is normally not supplied by the computer manufacturer. This software will permit the DBMS with all its computers and microprocessor controlled devices to function as a system. It includes the Integrated Data Base Management System, various special system processors such as the Packet Header Interface and the Configuration Management System. These software processors shall run within VAX/VMS at the user process level. It includes both procedure and data tables. The Integrated Data Base Management System is defined in paragraph 4.7.1. The Configuration Management System is defined in paragraph 4.7.2.

### 4.2.5 APPLICATIONS PROCESSES

The Applications Processes comprise the software specifically generated to manipulate data in the data base according to the needs of a specific user. The users may be mission, research, or operational. The software supplied by one class of users, for example a mission user, may be used by a different class of users as system specific software; however, it is still defined as application software because it is the responsibility of a user to provide it.

#### 4.2.5.1 Mission Applications Processes

Mission applications processes are the processes provided by mission users. An example is software that will function within the DBMS structure to reformat packetized data to forms compatible with established interface standards. The standards will be compatible with the DBMS quick-look and display software.

They will also provide a defined interface for the other user software. It is the responsibility of mission users to provide this software.

#### 4.2.5.2 Research Applications Processes

Research applications processes are the responsibility of and will be provided by the research user. It is not a design requirement of the DBMS to provide resources for the execution of these processors. Depending upon the operational policies, users may be permitted to execute their programs within the available system resources. There will be a limited provision for storing these programs and data for the convenience of the research user. In general, large research applications processes are not anticipated to be executed on the DBMS computers.

#### 4.2.5.3 Operational Applications Processes

Operational applications processes are those information extraction models and data files supplied by users with defined required outputs and scheduled requirements for data. It is not a design requirement of DBMS for these processes to execute on the DBMS computers. The DBMS shall have the capability to provide such data at a user terminal.

#### 4.2.5.4 Corollary Data

The utility of space acquired data is enhanced through corollary data. Operational applications users are one source of corollary data. The DBMS shall have the capability to manage such corollary data. However, responsibility for the data content and any required formatting software will be incumbent upon the user and not on the DBMS.

#### 4.2.5.5 Relational Tables

Relational tables defined by the application user are designated as "User Relational Tables." These are excluded from the Special Systems Software. Responsibility for these tables lies with the application user.

Relational tables defined by the data base administrator are designated as "System Relational Tables." They are a general nature required to operate the system and were included in the definition of Special Systems Software in paragraph 4.2.4.

### 4.3 VAX I OPERATING SYSTEM

The VAX I shall have a VAX/VMS operating system with the appropriate drivers for each connected peripheral device. These drivers shall include those normally supplied by Digital Equipment Company with the VAX/VMS for the standard peripherals and any special drivers required for DBMS unique devices. The operating system shall provide the environment for concurrent execution of multi-user timesharing, batch, and real-time applications. It shall provide the following:

- o virtual memory management for the execution of large programs
- o event-driven priority scheduling
- o shared memory, file, and interprocess communication data protection based on ownership and application groups
- o programmed system services for process and subprocess control and interprocess communication

#### 4.3.1 STANDARD DRIVERS

The operating system for VAX I shall include drivers for the following peripheral devices:

- o RP06      disc drive
- o TU45      75 ips magnetic tape transport
- o DZ-11A    Asynchronous multiplexer
- o LA36      DEC writer console terminal

#### 4.3.2 OTHER SERVICES

The VAX I operating system shall also provide for record management system (RMS) services, virtual I/O to logical file including mailboxes, and for access to the dual port memory shared with VAX II.

#### 4.3.3 AUXILIARY STORAGE DRIVER

The VAX I operating system shall include a driver to facilitate I/O transfers to the auxiliary storage device. This driver shall provide for byte and block data transfers via the UNIBUS to a disc controller supporting from 1 to 8 parallel transfer disc storage units. The driver shall also provide for the

control signals via the UNIBUS to accomplish reads and writes to the auxiliary storage via a second port on the controller.

#### 4.3.4 DATA BUS PORT DRIVER

The VAX I operating system shall include a data bus port driver to facilitate the I/O transfer of data between memory and the Data Bus Port via the UNIBUS. The driver shall comply with the following criteria:

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS and DR-11B
- c. provide for control signals via the UNIBUS to maintain protocol with the data bus port device on the DR-11B
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary data bus port specific subroutine and tables, and use existing VMS subroutines to perform the following functions:

- o define the data bus port for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the data bus port for operation at system start up and during recovery from a power failure
- o perform data bus port I/O preprocessing
- o translate programmed requests for I/O operations into commands that are specific to the data bus port
- o activate the data bus port
- o respond to interrupts from the data bus port
- o respond to time out conditions for the data bus port
- o respond to requests to cancel I/O on the data bus port
- o report data bus port errors to an error logging program
- o return status from the data bus port to the process that requested the I/O operation

#### 4.4 VAX II OPERATING SYSTEM

The VAX II shall have a VAX/VMS operating system with the appropriate drivers for each connected peripheral device. These drivers shall include those normally supplied by Digital Equipment Company with the VAX/VMS for the standard peripherals and any special drivers required for DRMS unique devices. The operating system shall provide the environment for concurrent execution of multi-user timesharing, batch, and real-time applications operating with a segment of dual ported memory shared with VAX I. It shall provide the following:

- o virtual memory management for the execution of large programs
- o event-driven priority scheduling
- o shared memory, file, and interprocess communication data protection based on ownership and application groups
- o programmed system services for process and subprocess control and interprocess communication.

##### 4.4.1 STANDARD DEVICES

The operating system for VAX II shall include drivers for the following devices:

- o RP06        disc drive
- o TE16        45 ips magnetic tape transport
- o DZ-11A      asynchronous multiplexer
- o LA120       DEC writer console terminal

##### 4.4.2 OTHER SERVICES

The VAX II operating system shall also provide for record management system (RMS) services, virtual I/O to logical file including mailboxes, and for access to the dual port memory shared with VAX I.

##### 4.4.3 MASTER CONTROLLER AND DATA BUS PORT DRIVER

The VAX II operating system shall include a Master Controller and Data Bus Port Driver to facilitate the I/O transfer of data between memory and the Master Controller and Data Bus Port via the UNIBUS and DR-11B. This driver shall have

control signals via the UNIBUS to accomplish reads and writes to the auxiliary storage via a second port on the controller.

#### 4.3.4 DATA BUS PORT DRIVER

The VAX I operating system shall include a data bus port driver to facilitate the I/O transfer of data between memory and the Data Bus Port via the UNIBUS. The driver shall comply with the following criteria:

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS and DR-11B
- c. provide for control signals via the UNIBUS to maintain protocol with the data bus port device on the DR-11B
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary data bus port specific subroutine and tables, and use existing VMS subroutines to perform the following functions:

- o define the data bus port for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the data bus port for operation at system start up and during recovery from a power failure
- o perform data bus port I/O preprocessing
- o translate programmed requests for I/O operations into commands that are specific to the data bus port
- o activate the data bus port
- o respond to interrupts from the data bus port
- o respond to time out conditions for the data bus port
- o respond to requests to cancel I/O on the data bus port
- o report data bus port errors to an error logging program
- o return status from the data bus port to the process that requested the I/O operation

#### 4.4 VAX II OPERATING SYSTEM

The VAX II shall have a VAX/VMS operating system with the appropriate drivers for each connected peripheral device. These drivers shall include those normally supplied by Digital Equipment Company with the VAX/VMS for the standard peripherals and any special drivers required for DRMS unique devices. The operating system shall provide the environment for concurrent execution of multi-user timesharing, batch, and real-time applications operating with a segment of dual ported memory shared with VAX I. It shall provide the following:

- o virtual memory management for the execution of large programs
- o event-driven priority scheduling
- o shared memory, file, and interprocess communication data protection based on ownership and application groups
- o programmed system services for process and subprocess control and interprocess communication.

##### 4.4.1 STANDARD DEVICES

The operating system for VAX II shall include drivers for the following devices:

- o RP06 disc drive
- o TE16 45 ips magnetic tape transport
- o DZ-11A asynchronous multiplexer
- o LA120 DEC writer console terminal

##### 4.4.2 OTHER SERVICES

The VAX II operating system shall also provide for record management system (RMS) services, virtual I/O to logical file including mailboxes, and for access to the dual port memory shared with VAX I.

##### 4.4.3 MASTER CONTROLLER AND DATA BUS PORT DRIVER

The VAX II operating system shall include a Master Controller and Data Bus Port Driver to facilitate the I/O transfer of data between memory and the Master Controller and Data Bus Port via the UNIBUS and DR-11B. This driver shall have

similar attributes and perform similar functions to the Data Bus Port Driver specified in paragraph 4.3.4. In addition, it shall accommodate the master controller functions. Principally, these require the assembly of control procedures for the master controller using tables, pre-established subroutines, and system specific data and software that is part of the special system software, and to output the resulting procedures and data to the proper memory locations in the master controller at the proper time. Timing will be asynchronously controlled from the master controller. This driver shall use other system specific software to aid in the execution of the unique master controller functions. The major system specific software will be the data bus traffic control and the data bus command processors as well as the command and configuration tables. These processors are described in paragraphs 4.7.3 through 4.7.3.4. The tables are described in paragraphs 4.7.5 through 4.7.5.4.

#### 4.4.4 IMAGE DISPLAY DRIVER

The VAX II operating system shall include an image display system driver to facilitate the I/O transfer of data between memory and the image display controller via the UNIBUS and DR11-B. The driver shall comply with the following criteria.

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS
- c. provide for control signals via the UNIBUS to maintain protocol with the display controller
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary display specific subroutine and tables and use existing VMS subroutines to perform the following functions:

- o define the image display for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the display controller for operation at system start up and during recovery from a power failure



similar attributes and perform similar functions to the Data Bus Port Driver specified in paragraph 4.3.4. In addition, it shall accommodate the master controller functions. Principally, these require the assembly of control procedures for the master controller using tables, pre-established subroutines, and system specific data and software that is part of the special system software, and to output the resulting procedures and data to the proper memory locations in the master controller at the proper time. Timing will be asynchronously controlled from the master controller. This driver shall use other system specific software to aid in the execution of the unique master controller functions. The major system specific software will be the data bus traffic control and the data bus command processors as well as the command and configuration tables. These processors are described in paragraphs 4.7.3 through 4.7.3.4. The tables are described in paragraphs 4.7.5 through 4.7.5.4.

#### 4.4.4 IMAGE DISPLAY DRIVER

The VAX II operating system shall include an image display system driver to facilitate the I/O transfer of data between memory and the image display controller via the UNIBUS and DR11-B. The driver shall comply with the following criteria.

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS
- c. provide for control signals via the UNIBUS to maintain protocol with the display controller
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary display specific subroutine and tables and use existing VMS subroutines to perform the following functions:

- o define the image display for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the display controller for operation at system start up and during recovery from a power failure

- o return status from the display controller and the displays to the process that requested the I/O operators

#### 4.5 SUPPORT PROCESSOR

The DBMS software shall include support processors as specified in the following paragraphs.

##### 4.5.1 STANDARD VAX/VMS SUPPORT PROCESSORS

The following support processors normally supplied with VAX/VMS shall be executable on either VAX I or VAX II.

- o Interactive Text Editor (SOS)
- o Batch Oriented Text Editor (SLP)
- o Linker
- o Debug
- o Common Run-Time Procedure Library
- o Record Management System (RMS)
- o Macro Assembler
- o Operator Communication
- o Command Interpreter

##### 4.5.2 OPTIONAL DEC SUPPORT PROCESSOR

The following support processors, available from DEC as options, shall be executable on either VAX I or VAX II:

- o VAX II FORTRAN IV Plus
- o Datatrieve
- o Bliss-32

##### 4.5.3 MICROCOMPUTER SUPPORT

The DBMS software shall include one or more cross assemblers and simulators that will execute on either VAX I or VAX II. Each microcomputer or other

control signals via the UNIBUS to accomplish reads and writes to the auxiliary storage via a second port on the controller.

#### 4.3.4 DATA BUS PORT DRIVER

The VAX I operating system shall include a data bus port driver to facilitate the I/O transfer of data between memory and the Data Bus Port via the UNIBUS. The driver shall comply with the following criteria:

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS and DR-11B
- c. provide for control signals via the UNIBUS to maintain protocol with the data bus port device on the DR-11B
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary data bus port specific subroutine and tables, and use existing VMS subroutines to perform the following functions:

- o define the data bus port for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the data bus port for operation at system start up and during recovery from a power failure
- o perform data bus port I/O preprocessing
- o translate programmed requests for I/O operations into commands that are specific to the data bus port
- o activate the data bus port
- o respond to interrupts from the data bus port
- o respond to time out conditions for the data bus port
- o respond to requests to cancel I/O on the data bus port
- o report data bus port errors to an error logging program
- o return status from the data bus port to the process that requested the I/O operation

#### 4.4 VAX II OPERATING SYSTEM

The VAX II shall have a VAX/VMS operating system with the appropriate drivers for each connected peripheral device. These drivers shall include those normally supplied by Digital Equipment Company with the VAX/VMS for the standard peripherals and any special drivers required for DRMS unique devices. The operating system shall provide the environment for concurrent execution of multi-user timesharing, batch, and real-time applications operating with a segment of dual ported memory shared with VAX I. It shall provide the following:

- o virtual memory management for the execution of large programs
- o event-driven priority scheduling
- o shared memory, file, and interprocess communication data protection based on ownership and application groups
- o programmed system services for process and subprocess control and interprocess communication.

##### 4.4.1 STANDARD DEVICES

The operating system for VAX II shall include drivers for the following devices:

- o RP06        disc drive
- o TE16        45 ips magnetic tape transport
- o DZ-11A      asynchronous multiplexer
- o LA120       DEC writer console terminal

##### 4.4.2 OTHER SERVICES

The VAX II operating system shall also provide for record management system (RMS) services, virtual I/O to logical file including mailboxes, and for access to the dual port memory shared with VAX I.

##### 4.4.3 MASTER CONTROLLER AND DATA BUS PORT DRIVER

The VAX II operating system shall include a Master Controller and Data Bus Port Driver to facilitate the I/O transfer of data between memory and the Master Controller and Data Bus Port via the UNIBUS and DR-11B. This driver shall have

similar attributes and perform similar functions to the Data Bus Port Driver specified in paragraph 4.3.4. In addition, it shall accommodate the master controller functions. Principally, these require the assembly of control procedures for the master controller using tables, pre-established subroutines, and system specific data and software that is part of the special system software, and to output the resulting procedures and data to the proper memory locations in the master controller at the proper time. Timing will be asynchronously controlled from the master controller. This driver shall use other system specific software to aid in the execution of the unique master controller functions. The major system specific software will be the data bus traffic control and the data bus command processors as well as the command and configuration tables. These processors are described in paragraphs 4.7.3 through 4.7.3.4. The tables are described in paragraphs 4.7.5 through 4.7.5.4.

#### 4.4.4 IMAGE DISPLAY DRIVER

The VAX II operating system shall include an image display system driver to facilitate the I/O transfer of data between memory and the image display controller via the UNIBUS and DR11-B. The driver shall comply with the following criteria.

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS
- c. provide for control signals via the UNIBUS to maintain protocol with the display controller
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary display specific subroutine and tables and use existing VMS subroutines to perform the following functions:

- o define the image display for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the display controller for operation at system start up and during recovery from a power failure

- o return status from the display controller and the displays to the process that requested the I/O operators

#### 4.5 SUPPORT PROCESSOR

The DBMS software shall include support processors as specified in the following paragraphs.

##### 4.5.1 STANDARD VAX/VMS SUPPORT PROCESSORS

The following support processors normally supplied with VAX/VMS shall be executable on either VAX I or VAX II.

- o Interactive Text Editor (SOS)
- o Batch Oriented Text Editor (SLP)
- o Linker
- o Debug
- o Common Run-Time Procedure Library
- o Record Management System (RMS)
- o Macro Assembler
- o Operator Communication
- o Command Interpreter

##### 4.5.2 OPTIONAL DEC SUPPORT PROCESSOR

The following support processors, available from DEC as options, shall be executable on either VAX I or VAX II:

- o VAX II FORTRAN IV Plus
- o Datatrieve
- o Bliss-32

##### 4.5.3 MICROCOMPUTER SUPPORT

The DBMS software shall include one or more cross assemblers and simulators that will execute on either VAX I or VAX II. Each microcomputer or other

similar attributes and perform similar functions to the Data Bus Port Driver specified in paragraph 4.3.4. In addition, it shall accommodate the master controller functions. Principally, these require the assembly of control procedures for the master controller using tables, pre-established subroutines, and system specific data and software that is part of the special system software, and to output the resulting procedures and data to the proper memory locations in the master controller at the proper time. Timing will be asynchronously controlled from the master controller. This driver shall use other system specific software to aid in the execution of the unique master controller functions. The major system specific software will be the data bus traffic control and the data bus command processors as well as the command and configuration tables. These processors are described in paragraphs 4.7.3 through 4.7.3.4. The tables are described in paragraphs 4.7.5 through 4.7.5.4.

#### 4.4.4 IMAGE DISPLAY DRIVER

The VAX II operating system shall include an image display system driver to facilitate the I/O transfer of data between memory and the image display controller via the UNIBUS and DR11-B. The driver shall comply with the following criteria.

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS
- c. provide for control signals via the UNIBUS to maintain protocol with the display controller
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary display specific subroutine and tables and use existing VMS subroutines to perform the following functions:

- o define the image display for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the display controller for operation at system start up and during recovery from a power failure

control signals via the UNIBUS to accomplish reads and writes to the auxiliary storage via a second port on the controller.

#### 4.3.4 DATA BUS PORT DRIVER

The VAX 1 operating system shall include a data bus port driver to facilitate the I/O transfer of data between memory and the Data Bus Port via the UNIBUS. The driver shall comply with the following criteria:

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS and DR-11B
- c. provide for control signals via the UNIBUS to maintain protocol with the data bus port device on the DR-11B
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary data bus port specific subroutine and tables, and use existing VMS subroutines to perform the following functions:

- o define the data bus port for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the data bus port for operation at system start up and during recovery from a power failure
- o perform data bus port I/O preprocessing
- o translate programmed requests for I/O operations into commands that are specific to the data bus port
- o activate the data bus port
- o respond to interrupts from the data bus port
- o respond to time out conditions for the data bus port
- o respond to requests to cancel I/O on the data bus port
- o report data bus port errors to an error logging program
- o return status from the data bus port to the process that requested the I/O operation



control signals via the UNIBUS to accomplish reads and writes to the auxiliary storage via a second port on the controller.

#### 4.3.4 DATA BUS PORT DRIVER

The VAX I operating system shall include a data bus port driver to facilitate the I/O transfer of data between memory and the Data Bus Port via the UNIBUS. The driver shall comply with the following criteria:

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS and DR-11B
- c. provide for control signals via the UNIBUS to maintain protocol with the data bus port device on the DR-11B
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary data bus port specific subroutine and tables, and use existing VMS subroutines to perform the following functions:

- o define the data bus port for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the data bus port for operation at system start up and during recovery from a power failure
- o perform data bus port I/O preprocessing
- o translate programmed requests for I/O operations into commands that are specific to the data bus port
- o activate the data bus port
- o respond to interrupts from the data bus port
- o respond to time out conditions for the data bus port
- o respond to requests to cancel I/O on the data bus port
- o report data bus port errors to an error logging program
- o return status from the data bus port to the process that requested the I/O operation

#### 4.4 VAX II OPERATING SYSTEM

The VAX II shall have a VAX/VMS operating system with the appropriate drivers for each connected peripheral device. These drivers shall include those normally supplied by Digital Equipment Company with the VAX/VMS for the standard peripherals and any special drivers required for DRMS unique devices. The operating system shall provide the environment for concurrent execution of multi-user timesharing, batch, and real-time applications operating with a segment of dual ported memory shared with VAX I. It shall provide the following:

- o virtual memory management for the execution of large programs
- o event-driven priority scheduling
- o shared memory, file, and interprocess communication data protection based on ownership and application groups
- o programmed system services for process and subprocess control and interprocess communication.

##### 4.4.1 STANDARD DEVICES

The operating system for VAX II shall include drivers for the following devices:

- o RP06        disc drive
- o TE16        45 ips magnetic tape transport
- o DZ-11A      asynchronous multiplexer
- o LA120       DEC writer console terminal

##### 4.4.2 OTHER SERVICES

The VAX II operating system shall also provide for record management system (RMS) services, virtual I/O to logical file including mailboxes, and for access to the dual port memory shared with VAX I.

##### 4.4.3 MASTER CONTROLLER AND DATA BUS PORT DRIVER

The VAX II operating system shall include a Master Controller and Data Bus Port Driver to facilitate the I/O transfer of data between memory and the Master Controller and Data Bus Port via the UNIBUS and DR-11B. This driver shall have

similar attributes and perform similar functions to the Data Bus Port Driver specified in paragraph 4.3.4. In addition, it shall accommodate the master controller functions. Principally, these require the assembly of control procedures for the master controller using tables, pre-established subroutines, and system specific data and software that is part of the special system software, and to output the resulting procedures and data to the proper memory locations in the master controller at the proper time. Timing will be asynchronously controlled from the master controller. This driver shall use other system specific software to aid in the execution of the unique master controller functions. The major system specific software will be the data bus traffic control and the data bus command processors as well as the command and configuration tables. These processors are described in paragraphs 4.7.3 through 4.7.3.4. The tables are described in paragraphs 4.7.5 through 4.7.5.4.

#### 4.4.4 IMAGE DISPLAY DRIVER

The VAX II operating system shall include an image display system driver to facilitate the I/O transfer of data between memory and the image display controller via the UNIBUS and DR11-B. The driver shall comply with the following criteria.

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS
- c. provide for control signals via the UNIBUS to maintain protocol with the display controller
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary display specific subroutine and tables and use existing VMS subroutines to perform the following functions:

- o define the image display for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the display controller for operation at system start up and during recovery from a power failure

- o return status from the display controller and the displays to the process that requested the I/O operators

#### 4.5 SUPPORT PROCESSOR

The DBMS software shall include support processors as specified in the following paragraphs.

##### 4.5.1 STANDARD VAX/VMS SUPPORT PROCESSORS

The following support processors normally supplied with VAX/VMS shall be executable on either VAX I or VAX II.

- o Interactive Text Editor (SOS)
- o Batch Oriented Text Editor (SLP)
- o Linker
- o Debug
- o Common Run-Time Procedure Library
- o Record Management System (RMS)
- o Macro Assembler
- o Operator Communication
- o Command Interpreter

##### 4.5.2 OPTIONAL DEC SUPPORT PROCESSOR

The following support processors, available from DEC as options, shall be executable on either VAX I or VAX II:

- o VAX II FORTRAN IV Plus
- o Datatrieve
- o Bliss-32

##### 4.5.3 MICROCOMPUTER SUPPORT

The DBMS software shall include one or more cross assemblers and simulators that will execute on either VAX I or VAX II. Each microcomputer or other

processor that uses programmable software shall be supported by at least one cross-assembler. Each device that uses field replaceable microcode or ROM components shall be supported by at least one simulator sufficient to permit the development of microcode in software prior to the commitment to firmware.

#### 4.5.3.1 Cross-Assembler Utilization

The cross-assemblers will be used to support the development and maintenance of the software programs for the target processors in the data bus ports. This is a necessary factor to assure future flexibility of the interfaces to changing displays, terminals, and user computers. The development and maintenance of system software on one of the system computers will provide access to the system utilities, file management, editors, and storage capacity of the DBMS.

#### 4.5.3.2 Simulator Utilization

The simulators will be used to verify logic prior to commitment to read only memory firmware and to support the development and maintenance of system specific software prior to the incorporation of the firmware into the system. The intent is to reduce the overall cost of microcode production. Therefore, simulation fidelity shall be only to the extent necessary and unnecessarily elaborate or costly simulation shall be avoided.

#### 4.5.4 MASTER CONTROLLER AND DATA BUS SIMULATOR

The master controller and data bus shall be simulated in software to the fidelity necessary to develop and support the maintenance of the data bus specific systems software. The simulator shall provide a table driven response of one or more status and data words to user processes or drivers in a manner like the operational device would respond. There shall be no requirement to maintain fidelity of timing with the simulator. Interface to the simulator in lieu of the device on the UNIBUS shall be implemented using the VAX/VMS utilities and system capability to redefine a virtual device so as to minimize any change to the process undergoing development.

#### 4.5.5 DATA BUS PORT SIMULATOR

The data bus port shall be simulated in software to the fidelity necessary to develop and support the maintenance of the data bus specific systems software. The simulator shall provide for table driven responses as well as interfaces to other device simulators in a manner like the operational data bus port would respond. There shall be no requirement to maintain fidelity of timing with the data bus port simulator. Interface to the simulator in lieu of the device on the UNIBUS shall be implemented using the VAX/VMS utilities and system capability to redefine a virtual device so as to minimize any changes to the process undergoing development.

#### 4.5.6 OTHER DEVICE SIMULATORS

Although not explicitly required, nothing in this specification shall preclude the incorporation into the support processors of software simulation for other system elements such as the archival mass memory, auxiliary storage or the staging area interface. Simulators shall be developed and used only when they will contribute to an overall reduction in the cost of developing and maintaining the system software.

#### 4.5.7 DISPLAY PROCESSORS

The DBMS shall have display software that will provide the proper sequence of embedded commands and data formatting for the image display controller. The input data format for this processor shall conform to the standard identified during the system design. The mission application processors will format the acquired data to conform to the identified standards.

### 4.6 DIAGNOSTICS

The DBMS software shall include diagnostics for the system computers and their standard peripheral devices and for the data bus port, the data bus, the archival mass memory, the auxiliary storage, the display system, and the staging area interface.

#### 4.6.1 COMPUTER AND STANDARD PERIPHERAL DEVICES

The diagnostics for the VAX 11/780's and their standard peripheral devices shall be those supplied with the VAX/VMS and the addition of the VAX 11/780 diagnostic extended package.

#### 4.6.2 DATA BUS PORT

The data bus port diagnostic shall exercise the data bus port interfaced to a system computer, provide a positive indication of proper performance and a first level indication of the probable malfunction. The diagnostic shall provide for the automatic sequence of command and data outputs and inputs as well as the exercise of a single command or function of the device. The diagnostic shall provide for the automatic continual repetition of either the entire sequence or individual steps.

#### 4.6.3 DATA BUS

The data bus diagnostic shall exercise the master controller and data bus port interfaced to a system computer and each data bus port on the data bus. It shall provide a positive indication of proper performance and a first level indication of the probable malfunction. The diagnostic shall provide for the automatic sequence of command and data outputs and inputs as well as the exercise of a single command or function of each of the devices. The diagnostic shall provide for the automatic continual repetition of either the entire sequence or individual steps.

#### 4.6.4 ARCHIVAL MASS MEMORY

The archival mass memory diagnostic shall be limited to verifying the proper response at the AMM interface to a data bus port. As a minimum, it shall provide the capability for a direct operator designation of a packet according to the prescribed identification code and the subsequent commands to retrieve and verify the retrieval of the identified data packet.

#### 4.6.5 AUXILIARY STORAGE

The auxiliary storage diagnostic shall exercise the UNIBUS interface, the data bus port interface, the controller, and each of the discs. It shall provide

control signals via the UNIBUS to accomplish reads and writes to the auxiliary storage via a second port on the controller.

#### 4.3.4 DATA BUS PORT DRIVER

The VAX 1 operating system shall include a data bus port driver to facilitate the I/O transfer of data between memory and the Data Bus Port via the UNIBUS. The driver shall comply with the following criteria:

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS and DR-11B
- c. provide for control signals via the UNIBUS to maintain protocol with the data bus port device on the DR-11B
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary data bus port specific subroutine and tables, and use existing VMS subroutines to perform the following functions:

- o define the data bus port for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the data bus port for operation at system start up and during recovery from a power failure
- o perform data bus port I/O preprocessing
- o translate programmed requests for I/O operations into commands that are specific to the data bus port
- o activate the data bus port
- o respond to interrupts from the data bus port
- o respond to time out conditions for the data bus port
- o respond to requests to cancel I/O on the data bus port
- o report data bus port errors to an error logging program
- o return status from the data bus port to the process that requested the I/O operation



#### 4.4 VAX 11 OPERATING SYSTEM

The VAX 11 shall have a VAX/VMS operating system with the appropriate drivers for each connected peripheral device. These drivers shall include those normally supplied by Digital Equipment Company with the VAX/VMS for the standard peripherals and any special drivers required for DBMS unique devices. The operating system shall provide the environment for concurrent execution of multi-user timesharing, batch, and real-time applications operating with a segment of dual ported memory shared with VAX 1. It shall provide the following:

- o virtual memory management for the execution of large programs
- o event-driven priority scheduling
- o shared memory, file, and interprocess communication data protection based on ownership and application groups
- o programmed system services for process and subprocess control and interprocess communication.

##### 4.4.1 STANDARD D.E.

The operating system for VAX 11 shall include drivers for the following devices:

- o RP06        disc drive
- o TE16        45 ips magnetic tape transport
- o DZ-11A      asynchronous multiplexer
- o LA120       DEC writer console terminal

##### 4.4.2 OTHER SERVICES

The VAX 11 operating system shall also provide for record management system (RMS) services, virtual I/O to logical file including mailboxes, and for access to the dual port memory shared with VAX 1.

##### 4.4.3 MASTER CONTROLLER AND DATA BUS PORT DRIVER

The VAX 11 operating system shall include a Master Controller and Data Bus Port Driver to facilitate the I/O transfer of data between memory and the Master Controller and Data Bus Port via the UNIBUS and DR-11B. This driver shall have

similar attributes and perform similar functions to the Data Bus Port Driver specified in paragraph 4.3.4. In addition, it shall accommodate the master controller functions. Principally, these require the assembly of control procedures for the master controller using tables, pre-established subroutines, and system specific data and software that is part of the special system software, and to output the resulting procedures and data to the proper memory locations in the master controller at the proper time. Timing will be asynchronously controlled from the master controller. This driver shall use other system specific software to aid in the execution of the unique master controller functions. The major system specific software will be the data bus traffic control and the data bus command processors as well as the command and configuration tables. These processors are described in paragraphs 4.7.3 through 4.7.3.4. The tables are described in paragraphs 4.7.5 through 4.7.5.4.

#### 4.4.4 IMAGE DISPLAY DRIVER

The VAX II operating system shall include an image display system driver to facilitate the I/O transfer of data between memory and the image display controller via the UNIBUS and DR11-B. The driver shall comply with the following criteria.

- a. be compatible with established VMS I/O protocol
- b. be compatible with the UNIBUS
- c. provide for control signals via the UNIBUS to maintain protocol with the display controller
- d. provide for three modes of transfer: programmed direct, DMA direct, and DMA buffered

The driver shall contain the necessary display specific subroutine and tables and use existing VMS subroutines to perform the following functions:

- o define the image display for the rest of VAX/VMS
- o define the driver for the system procedure that loads the driver into the system virtual address space and that creates its associated data structures
- o ready the display controller for operation at system start up and during recovery from a power failure

processor that uses programmable software shall be supported by at least one cross-assembler. Each device that uses field replaceable microcode or ROM components shall be supported by at least one simulator sufficient to permit the development of microcode in software prior to the commitment to firmware.

#### 4.5.3.1 Cross-Assembler Utilization

The cross-assemblers will be used to support the development and maintenance of the software programs for the target processors in the data bus ports. This is a necessary factor to assure future flexibility of the interfaces to changing displays, terminals, and user computers. The development and maintenance of system software on one of the system computers will provide access to the system utilities, file management, editors, and storage capacity of the DBMS.

#### 4.5.3.2 Simulator Utilization

The simulators will be used to verify logic prior to commitment to read only memory firmware and to support the development and maintenance of system specific software prior to the incorporation of the firmware into the system. The intent is to reduce the overall cost of microcode production. Therefore, simulation fidelity shall be only to the extent necessary and unnecessarily elaborate or costly simulation shall be avoided.

#### 4.5.4 MASTER CONTROLLER AND DATA BUS SIMULATOR

The master controller and data bus shall be simulated in software to the fidelity necessary to develop and support the maintenance of the data bus specific systems software. The simulator shall provide a table driven response of one or more status and data words to user processes or drivers in a manner like the operational device would respond. There shall be no requirement to maintain fidelity of timing with the simulator. Interface to the simulator in lieu of the device on the UNIBUS shall be implemented using the VAX/VMS utilities and system capability to redefine a virtual device so as to minimize any change to the process undergoing development.

#### 4.5.5 DATA BUS PORT SIMULATOR

The data bus port shall be simulated in software to the fidelity necessary to develop and support the maintenance of the data bus specific systems software. The simulator shall provide for table driven responses as well as interfaces to other device simulators in a manner like the operational data bus port would respond. There shall be no requirement to maintain fidelity of timing with the data bus port simulator. Interface to the simulator in lieu of the device on the UNIBUS shall be implemented using the VAX/VMS utilities and system capability to redefine a virtual device so as to minimize any changes to the process undergoing development.

#### 4.5.6 OTHER DEVICE SIMULATORS

Although not explicitly required, nothing in this specification shall preclude the incorporation into the support processors of software simulation for other system elements such as the archival mass memory, auxiliary storage or the staging area interface. Simulators shall be developed and used only when they will contribute to an overall reduction in the cost of developing and maintaining the system software.

#### 4.5.7 DISPLAY PROCESSORS

The DBMS shall have display software that will provide the proper sequence of embedded commands and data formatting for the image display controller. The input data format for this processor shall conform to the standard identified during the system design. The mission application processors will format the acquired data to conform to the identified standards.

### 4.6 DIAGNOSTICS

The DBMS software shall include diagnostics for the system computers and their standard peripheral devices and for the data bus port, the data bus, the archival mass memory, the auxiliary storage, the display system, and the staging area interface.

#### 4.6.1 COMPUTER AND STANDARD PERIPHERAL DEVICES

The diagnostics for the VAX 11/780's and their standard peripheral devices shall be those supplied with the VAX/VMS and the addition of the VAX 11/780 diagnostic extended package.

#### 4.6.2 DATA BUS PORT

The data bus port diagnostic shall exercise the data bus port interfaced to a system computer, provide a positive indication of proper performance and a first level indication of the probable malfunction. The diagnostic shall provide for the automatic sequence of command and data outputs and inputs as well as the exercise of a single command or function of the device. The diagnostic shall provide for the automatic continual repetition of either the entire sequence or individual steps.

#### 4.6.3 DATA BUS

The data bus diagnostic shall exercise the master controller and data bus port interfaced to a system computer and each data bus port on the data bus. It shall provide a positive indication of proper performance and a first level indication of the probable malfunction. The diagnostic shall provide for the automatic sequence of command and data outputs and inputs as well as the exercise of a single command or function of each of the devices. The diagnostic shall provide for the automatic continual repetition of either the entire sequence or individual steps.

#### 4.6.4 ARCHIVAL MASS MEMORY

The archival mass memory diagnostic shall be limited to verifying the proper response at the AMM interface to a data bus port. As a minimum, it shall provide the capability for a direct operator designation of a packet according to the prescribed identification code and the subsequent commands to retrieve and verify the retrieval of the identified data packet.

#### 4.6.5 AUXILIARY STORAGE

The auxiliary storage diagnostic shall exercise the UNIBUS interface, the data bus port interface, the controller, and each of the discs. It shall provide

a positive indication of proper performance and a first level indication of the probable malfunction. It shall provide for the independent and the combined operation of each of the specified elements. The diagnostic shall provide for the automatic sequence of command and data outputs and inputs as well as the exercise of a single command or function of the devices. The diagnostic shall provide for the automatic continual repetition of either the entire sequence or individual steps.

#### 4.6.6 DISPLAY SYSTEM

The display system diagnostics shall exercise the display controller, displays, and interfaces. It shall provide for the automatic transfer of established test data and for the transfer of manual test data designated by the operator.

#### 4.6.7 STAGING AREA INTERFACE

The staging area interface diagnostics shall provide for the independent exercise of the staging area interface, the header data bus port, the instrument packet data bus port, and the combined operation of all three segments. It is recognized that the staging area interface is an output device that is not amenable to stimulation via the system computer. The diagnostics may be limited to the necessary support and monitoring functions that will verify the proper function of these elements when subject to external manual stimulation, either by means of a hardware simulator or incorporated microcode self-test functions.

#### 4.7 SPECIAL SYSTEM SOFTWARE

The DBMS shall have the special systems software as described in the following paragraphs. The major classification of the routines, programs, algorithms, data, processors, and subsystem shall be: Integrated Data Base Management System (IDBMS), Configuration Management System (CMS), System Data Tables, and Demonstration System. These systems are described in paragraphs 4.7.1 through 4.7.4, respectively. The IDBMS will be furnished by the government.

#### 4.7.1 INTEGRATED DATA BASE MANAGEMENT SYSTEM

The IDBMS is comprised of three major subsystems. They are 1) the Data Base Processor, 2) the Data File Processor, and 3) the System Control Processor. These subsystems operate asynchronously at the user process level within VAX 1 computer under the VAX/VMS operating system. The Data Base Processor provides the user and application program an interface to data via relational tables. The data tables themselves are excluded from the class of special system software. The Data File Processor provides the interface to the data stored in the DBMS. The data itself is not included in the special system software. The System Control Processor is described with other processors in paragraph 4.7.1.3.

##### 4.7.1.1 Data Base Processor

The Data Base Processor provides a multi-user interface to the structured data sets. It will handle interactive and batch users and other applications processes. The DBP shall include inherent processes and shall utilize other VMS system processes to schedule, initiate, and suspend processes pending the asynchronous execution of external processes. It supports an interactive query language and provides for independent user relational data tables. It will identify data in the data base to file or sub-file level.

##### 4.7.1.2 Data File Processor

The Data File Processor provides the interface between the file or sub-file identification, provided by the Data Base Processor, and the data stored or archived in the DBMS. The data may be at the packet level in the archival mass memory or stored at user terminals. The DFP maintains the necessary cognizance of the physical location of the desired data so it can direct the retrieval commands to the Configuration Management System. This will include an identification of data archived in the AMM to a unique packet code. This code will be 56 bits long. Additionally, two 32-bit fields will identify the starting byte within the packet and the number of bytes requested. For files requiring multiple packets to complete the data file, the DFP will identify each of the required packets. A continuous sequence of packets may be identified by the starting packet and ending packet.

#### 4.7.1.3 Other Processors

In addition to the Data Base and the Data File Processors, the IDBMS shall contain system control processors and other utility processors necessary to assure the performance of the system. As a minimum, there shall be a Packet Header Interface which shall provide for the designations of relational tables according to the content of the data headers, other tabular data supplied by the project, and utility packets. The Packet Header Interface shall construct relational tables with a minimum of manual intervention using pre-established algorithms and the contents of the packet header data.

#### 4.7.2 CONFIGURATION MANAGEMENT SYSTEM

The Configuration Management System is that portion of the special systems software that is not included in the Integrated Data Base Management Software. It shall consist of processors executable in the systems computers and routines executable in microcomputers distributed throughout the systems devices. It will execute predominantly on VAX 11. It includes the minimum amounts of systems relational data tables to exercise the system to verify that it is functioning properly. The system shall provide for the principal functions of the DBMS: routing of incoming data for storage in the AMM, retrieval of archived data as directed by the IDBMS, the routing and control of data flow within the system, and the generation and manipulation of data to aid its retrieval and delivery to users.

##### 4.7.2.1 System Approach

The approach used to implement the configuration management functions shall be to employ table driven software to ensure flexibility and future expansion of system capability. This will also provide the flexibility for a phased implementation, should it be necessary, of initially executing all the processes on VAX 1 and subsequently separating them to execute on VAX 1 and VAX 11. The various system processors shall provide a hierarchical cognizance of functions and status of each system device down to the detailed commands and execution steps for functions within each device. The major processors are specified in paragraphs 4.7.3 through 4.7.4.4. The major tables are specified in paragraph 4.7.5, System Data Tables.



#### 4.7.2.2 Processor and Table Interaction

The principal function of the CMS software is illustrated in Figure 4-1. The Data Bus Configuration Processor functions asynchronously to maintain the status of the system configuration. The Data Bus Traffic Control Processor provides the interface to the rest of the world. It accepts commands and through the use of the Data Bus Configuration Table and the Command Table, directs control to the proper sequence of command processes. These processes utilize information in the command table and the devices tables for the devices involved for the execution of commands. These processes construct the proper words in memory for subsequent use by the Data Bus Traffic Control Processor and the Master Controller Driver. After completion of the external execution of the command, the Data Bus Traffic Control Processor performs some post processing to complete the communication with the external system. Most of the external system commands will originate with the IDBMS.

#### 4.7.3 MINICOMPUTER PROCESSORS

The principal processors of the CMS shall operate in VAX 11. Those processors are: the Data Bus Configuration Processor, the Data Bus Traffic Control Processor, the Data Bus Command Processors, the Data Bus Device Processors, and the Error Logging Processor. The Auxiliary Storage Housekeeping Processor will operate in VAX 1.

##### 4.7.3.1 Data Bus Configuration Processor

The Data Bus Configuration Processor (DBCP) shall maintain the configuration profile of the DBMS including the device assignments to data bus ports, their attributes, and activity status. This processor shall provide the coordination among device specific processors. It shall use the system data table defined in paragraph 4.7.5.

##### 4.7.3.2 Data Bus Traffic Control Processor

The Data Bus Traffic Control Processor (DBTCP) shall accept commands from external systems, perform the scheduling of data flow on the data bus, maintain queues of pending data transfers, and function to maximize the data throughput. Its primary function, in addition to maintaining queues of data transfer, shall be to resolve any potential conflict between data bus traffic and the staging

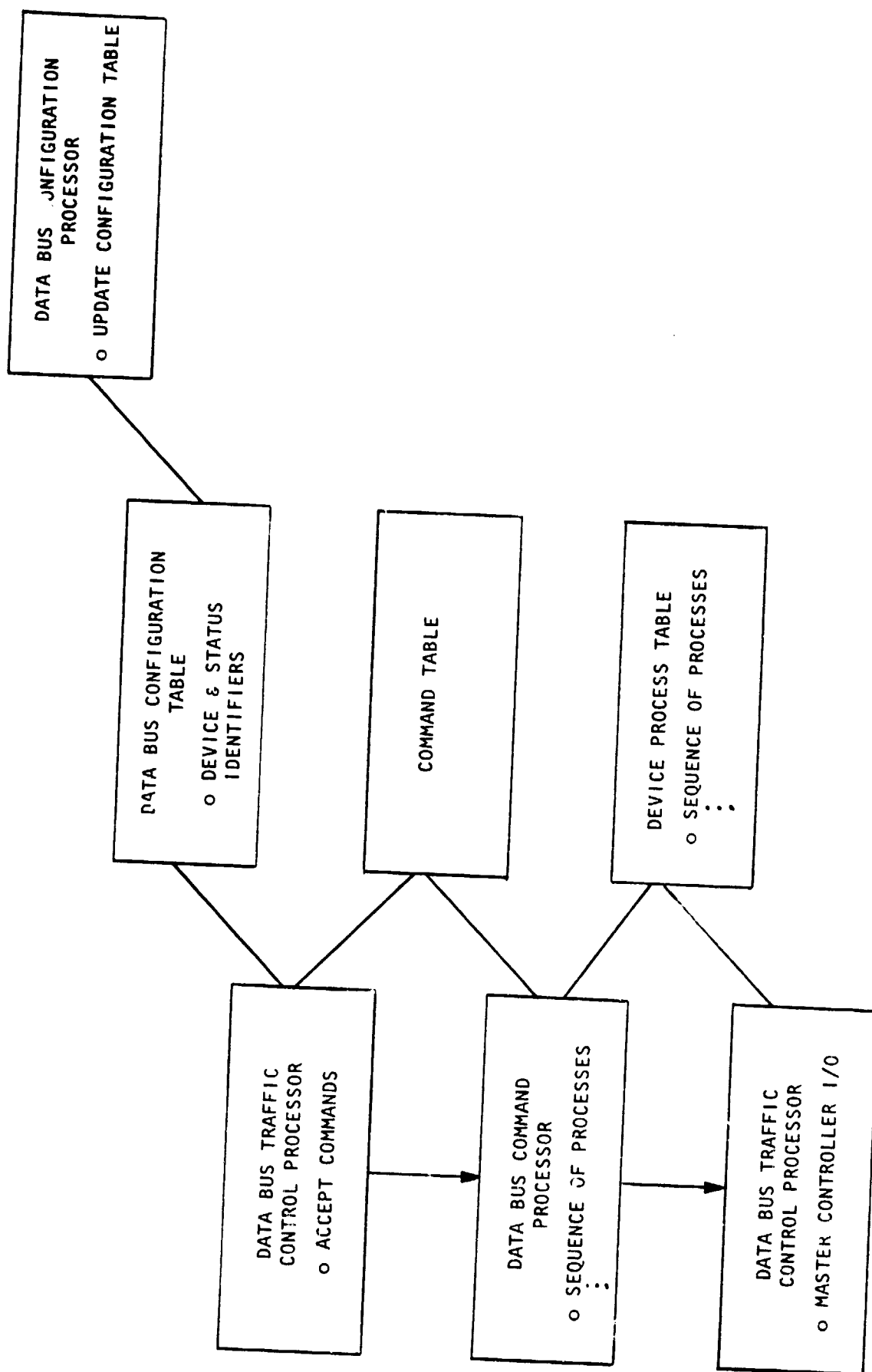


Figure 4-1. Data Bus Processing Overview

area interface for writing access to the archival mass memory. Adequate time to schedule archival storage activities is expected during the protocol for initiating data transfers from the staging area. Staging area data shall have priority over internal data bus traffic. Prior to scheduling a data transfer to the AMM, the DBTCP shall check the status of the staging area interface to ascertain that it is not engaged in the receipt of any messages from the staging area. If the staging area is active, no Class II data transfers shall be scheduled to the AMM. A twenty percent duty cycle is anticipated at the staging area interface.

#### 4.7.3.3 Data Bus Command Processor

The Command Processor (CP) shall be a set of processes, each executable according to the function or command to be performed by the DBMS. For example, the function of retrieving data from AMM and transferring it to a "device" or "port A" would call for the execution of the retrieval process. It in turn would use the AMM device process as well as the master controller device process and the Data Bus Traffic Control Process. The specific identification of the required sequence of processes, the data, and the command words will be obtained from the system tables.

#### 4.7.3.4 Data Bus Device Processors

The Device Processors shall be a set of processes, each executable according to the specific requirements of the connected device. The set of processes to be executed shall be determined by the Data Bus Command Processor from the Device Processor Table. The processes required to support the data bus port devices, such as the AMM, the user terminals, the master controller, and the staging area interface shall function similar to a driver in VMS except in the DBMS, the devices are physically removed from the host computer via the data bus. Additionally, some of the processes may execute in microcomputers that are a part of the data bus ports.

As a minimum, there shall be a process or a set of processes for each of the following devices:

- o Archival Mass Memory
- o Auxiliary Storage

- o User Terminal (Image Display)
- o User Terminal (Multiplexed)
- o User Terminal (Auxiliary multiplexer)
- o User Terminal (A/N Display)
- o Staging Area Interface Header
- o Staging Area Interface Packet Data
- o VAX I DR11-B
- o VAX II DR11-B

There shall be user terminal processors to provide for the execution of the proper sequence of utilities, according to the devices identified in the data bus configuration tables and the data bus ports identified for the data transfers. There shall be as many display processors as required to satisfy the devices provided in the system configuration.

#### 4.7.3.5 Error Logging Processor

The error logging processor shall be notified of errors by the error routines of the device drivers and other processors. It shall be a rudimentary processor with provisions by means of tables to be expandable to more detailed error reporting and fault isolation. As a minimum, it shall report on data transfer failures resulting from either time out or an excess number of unsuccessful transfer attempts on the data bus without acknowledgement. The error report shall identify the data bus ports involved.

#### 4.7.3.6 Auxiliary Storage Housekeeping Processor

The Auxiliary Storage Housekeeping Processor (ASHP) shall maintain the directory of auxiliary storage space, provide for any dynamic allocation of space as a supporting resource such as for staging large data sets during reformatting, and shall initiate file purge and compression when notified of the condition that packet headers are no longer needed. The Header Data Manager identified in paragraph 2.9.4 forms the major subset of this processor.

#### 4.7.4 MICROPROCESSOR ROUTINES

Each transmitter or receiver port interfaced to the fiber optics data bus will contain a programmable device, hereafter called a microprocessor. Software in the form of computer programs, routines, and microcode shall be provided for each of these DBMS elements as required to perform their specified functions. Specific device dependent software requirements are identified in the following paragraphs.

##### 4.7.4.1 Staging Area Interface Microprocessor Routines

The staging area interface will have one or more programmable devices. Programs shall be provided to perform the functions of transferring the data packets to the archival mass memory and the headers to the designated locations in auxiliary storage. Microprocessor software shall also supplement the hardwired logic in the performance of the X25 L4 protocol and supervisory packet generation and decoding.

##### 4.7.4.2 AMM Interface Microprocessor Port

The AMM interface port will have one or more programmable devices. Programs shall be provided to interpret and execute the data bus control functions, including data transfers to and from the data bus.

##### 4.7.4.3 Master Controller Microprocessor

The master controller and data bus port will have one or more programmable devices to initiate and control data transfers on the data bus. The master controller routines shall interface with supporting tables and processes operating in the system computer (VAX 11). The master controller, including the software programs, shall perform the following functions:

- o Maintain status of connected data ports
- o Insert the proper sequences of commands during command time
- o Output the proper poll, and acknowledge signals during the synchronization time
- o Respond to interrupts
- o Detect and initiate response to error conditions on the data bus

- o Transfer status and data words to the configuration management system processors
- o Receive compiled control words from the Configuration Management (VAX II) system processor
- o Decode and respond to control words received from the Configuration Management system processor

#### 4.7.4.4 Data Bus Ports Microprocessor

Each of the data bus ports, primarily VAX I and user terminals, will contain a programmable device. Each port shall have routines that shall be executable on the designated device to interpret and execute the command words, provide data bus interfacing protocol, and to provide the protocol to the interfacing device.

#### 4.7.5 SYSTEM DATA TABLES

The DBMS system is structured to permit demonstration of full system operational performance with a minimal system configuration. A table driven approach is employed toward this end. The data content of these tables that are used by the system is included in this definition. An example of such a table is a pre-established sequence of command words that are used to effect a data bus communication. As a minimum, the CMS shall contain the tables described in the following paragraphs.

##### 4.7.5.1 Data Bus Configuration Table

The Data Bus Configuration Table (DBCT) shall provide the specific data to tie the DBMS together as a system. The table shall provide the cross reference between the device codes referenced within the software processors and the data bus ports. It shall also identify the physical device attached to the port along with the characteristics of each. Characteristics shall include, but not be limited to, priority, sequential or random access to input or output device, Class I or Class II type data, and the flexibility of the device. Additional data, such as expected response times, shall be provided for, as it may be incorporated in future scheduling algorithms. Provisions shall be provided for in the table for future statistics such as the mean number of blocks transferred for any message.

The DBCT shall include the status of each data port and connected device. This status shall include, but not be limited to, busy, available, standby awaiting additional control, transfer in processes, disabled due to device failure, and awaiting device response.

The implementation of the above tabular function shall be systematic. Nothing in the preceding paragraph is intended to restrict the implementation to a single physical table.

#### 4.7.5.2 Device Process Table

The Device Process Table (DPT) shall provide the identification of device unique processes available to or required for the execution of a command on a given data port. Access to the table for the appropriate sequence of processes shall be dependent upon the device characteristics and status and the functions being executed. These functions may be identified in other processors, including user applications processors, or from data bus commands. The data bus commands shall comprise the total function code and the argument. The command processor may access the device access table using a combination of the commands and subsequent data words.

#### 4.7.5.3 Command Process Table

The Command Process Table shall provide the linkage for the command processor and the specific processes to be executed. The Command Process Table shall include the necessary assemblage of binary codes to construct sequences of command words for output over the data bus to the devices and the data bus ports. As a minimum, the table shall contain the following commands as listed in Table 4-1, and the sequence of commands required to execute each of them.

#### 4.7.5.4 Auxiliary Storage Map

The CMS shall contain a map of the auxiliary storage. This map shall contain the current status of dynamically allocated space as well as pre-established allocations for specific contexts.

Table 4-1. Bus Function Codes

- o Prepare to transmit packet from staging area
- o Prepare to receive packet from staging area
- o Prepare to transmit packet from DBMS
- o Prepare to receive packet from DBMS
- o Decode operand for format information
- o Pass through operand for device control
- o Decode operand for device control
- o Respond to polling query
- o Acknowledge receipt of packet
- o Request for retransmission
- o Prepare to transmit block of data
- o Prepare to receive block of data
- o Abort packet transfer
- o No operation
- o Return test data
- o Test data returned
- o Establish User Terminal Communication

#### 4.7.6 VERIFICATION SOFTWARE

There shall be a demonstration procedure and data tables that will exercise each of the following DBMS functions and provide an indication of proper performance.

- o Storage of a multiple block data set in archival mass memory
- o Retrieval of a multiple block data set from archival mass memory
- o The generation of a multiple block data set and a unique identification code



- o The execution of the IDBMS from a user terminal for archived data sets
- o The timing of simultaneous Class I and Class II data transfers on the data bus to indicate at least a 50 megabit transfer rate for each data class

#### 4.7.6.1 Fiber Optics Data Transfer Rate Verification

The verification of the simultaneous Class I and Class II data transfers on the fiber optics data bus may be accomplished by a combination of high speed data transfers from the data bus port buffers without requiring transfers to the external devices. The combined read and write capability of the Archival Mass Memory may also be used for this verification. The following approach is suggested to verify the capability of each port on the data bus to transmit and receive its prescribed rate.

#### 4.7.6.2 Test Conditions

The AMM shall have two files "1" and "2" identified. File "1" shall be zeroed. File "2" shall have a pre-established set of unique 2048 bit words. They are called "K" through "XXX."

Each Staging Area Interface Port shall have their buffers loaded with unique 2048 bit words. The header buffer shall be designated word "A" and the Packet Buffer shall be words "B" through "I."

The AMM ports and the ports of the initial transfers shall be properly set up for local microcomputer control.

A special command code shall be designated which will be interrupted by a port to receive a block and then transmit the same block out of the buffer at the next poll signal.

The VAX II port buffer shall be loaded with a unique word "J."

#### 4.7.6.3 Data Bus Transfers

The sequence of transfers identified below shall take place. Class I and Class II transfers shall occur concurrently. For simplification, the following abbreviations are used:

SAA	Staging Area Header Buffer Word A
SAB	Staging Area Packet Buffer Word B
SAC (etc)	Staging Area Packet Buffer Word C (etc)
VAX II C	Command word from VAX II buffer
VAX I	VAX I Port Buffer
AS	Auxiliary Storage Buffer
UTn	User Terminal n
AMM as transmitter	file 2
AMM as receiver	file 1
VAX II	VAX II Port Buffer with Data Word "J"

The following sequence shall transpire:

<u>Class I Transfer</u>	<u>Class II Transfer</u>
VAX II to AMM	AMM to VAX I
SAA to AMM	AMM to UT1
SAB to AMM	AMM to UT2
SAC to AMM	
SAD to AMM	
SAE to AMM	VAX II C to AS
.	
SAI to AMM	
VAX I to AMM	AMM to AS
SAB to AMM	

Class I Transferr

UT1 to AMM

UT2 to AMM

SAB to AMM

SAC to AMM

SAD to AMM

VAX I to AMM

SAE to AMM

.

.

Class II Transfer

VAX II C to VAX I

AMM to VAX I

VAX II C to UT1

AMM to UT1

The sequence of transfers shall be designed to test each data bus port and to require worst case data bus loading. Because of the need to exercise more ports transmitting to the AMM than receiving data from them, not all Class II times must be filled. Care should be exercised in designing the test sequence to bunch Class II transfers whenever sufficient receiving ports are available. Turnaround time of 50 microseconds is permissible to establish the control words in the Master Controller. Also, turnaround time of 30 microseconds is permissible for any port to decode the command word.

At the completion of the test, which shall be clocked using the VAX timer, the results of file 1 on the AMM shall be displayed and compared with the expected results.

#### 4.8 SOFTWARE OVERVIEW

The relationship of the DBMS software is illustrated in Figure 4-1. It shall operate on 1 or 2 VAX computers plus the microcomputers. The microcomputer software is shown outside the circle of Figure 4-2. The software operating within the VAX computers will be identical whether it is in one machine or two with shared memory. The VAX is a virtual machine with distinct boundaries between processors. Interprocess communications will be via mailboxes and common.

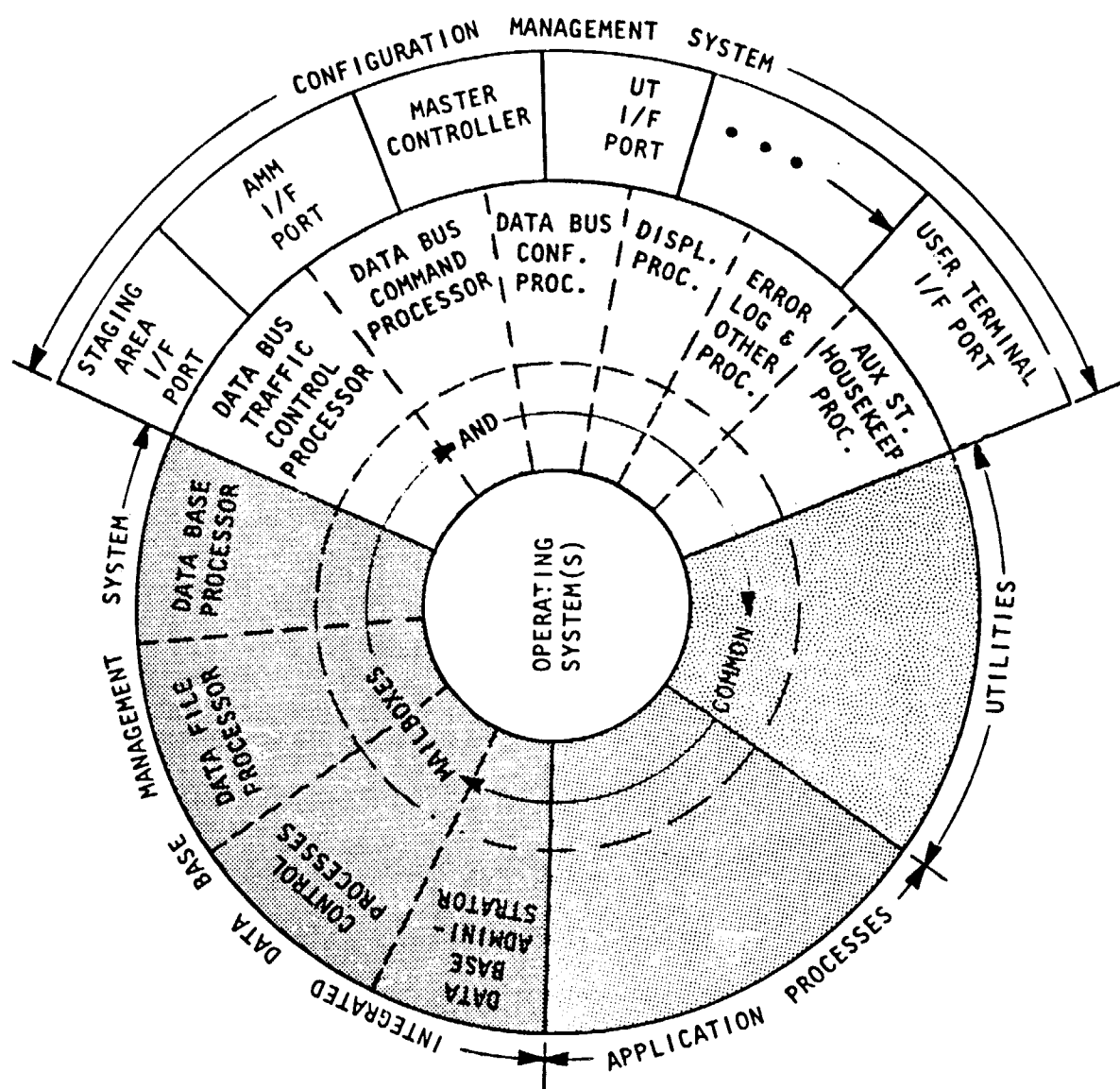


Figure 4-2. DBMS Software

The Integrated Data Base Management is the GSFC supplied software system. It is designated for execution on VAX I. The Configuration Management System is the DBMS specific system software, of which all the processors except the auxillary storage housekeeping are designated for execution on VAX II.

#### 4.8.1 TYPICAL DATA FLOW FROM DBMS TO RETRIEVE AMM DATA

The data flow shown in Figure 4-3 is typical of the interprocess and inter-device interactions within the DBMS. For this example, an I/O read is to be performed within the Data File Processor executing in VAX I. The device channel is assigned to the Data Bus Traffic Control Processor (DBTCP) mail box in the shared memory space.

The VMS performs the read operation by awakening the necessary mail box drivers, placing the message in the mail box, and awakening the DBTCP. The DBTCP reads the message and determines a command is present. It accesses the Data Bus Control Table (DBCT) to identify the devices and data ports required to execute the command. For this example, the DBCT directs the sequence to the Archival Mass Memory Process Table (AMMPT). This is a comparable process to that performed by any VMS system I/O driver. The sequence of processes identified by the AMMPT will be executed. Some of the processes may be VMS level fork processes; others may be CMS full processes and others may be AMM specific. The last process in the sequence will contain the memory locations in its space in which the control and data words necessary for the retrieval of the data from the AMM were assembled by the previously executed processes.

This last process will perform an I/O, or a sequence of I/O to the Data Bus Port and Master Controller (MC) using the MC driver installed in the VMS.

The MC driver will direct the proper sequence of commands over the UNIBUS and the DR-11B to effect communication with the MC. The transfer in the VAX II is now under UNIBUS control, effectively freeing the computer for other processes.

The MC will load its memory with the commands assembled in the VAX memory, decode some of them and initiate a command transfer to the AMM port. The MC

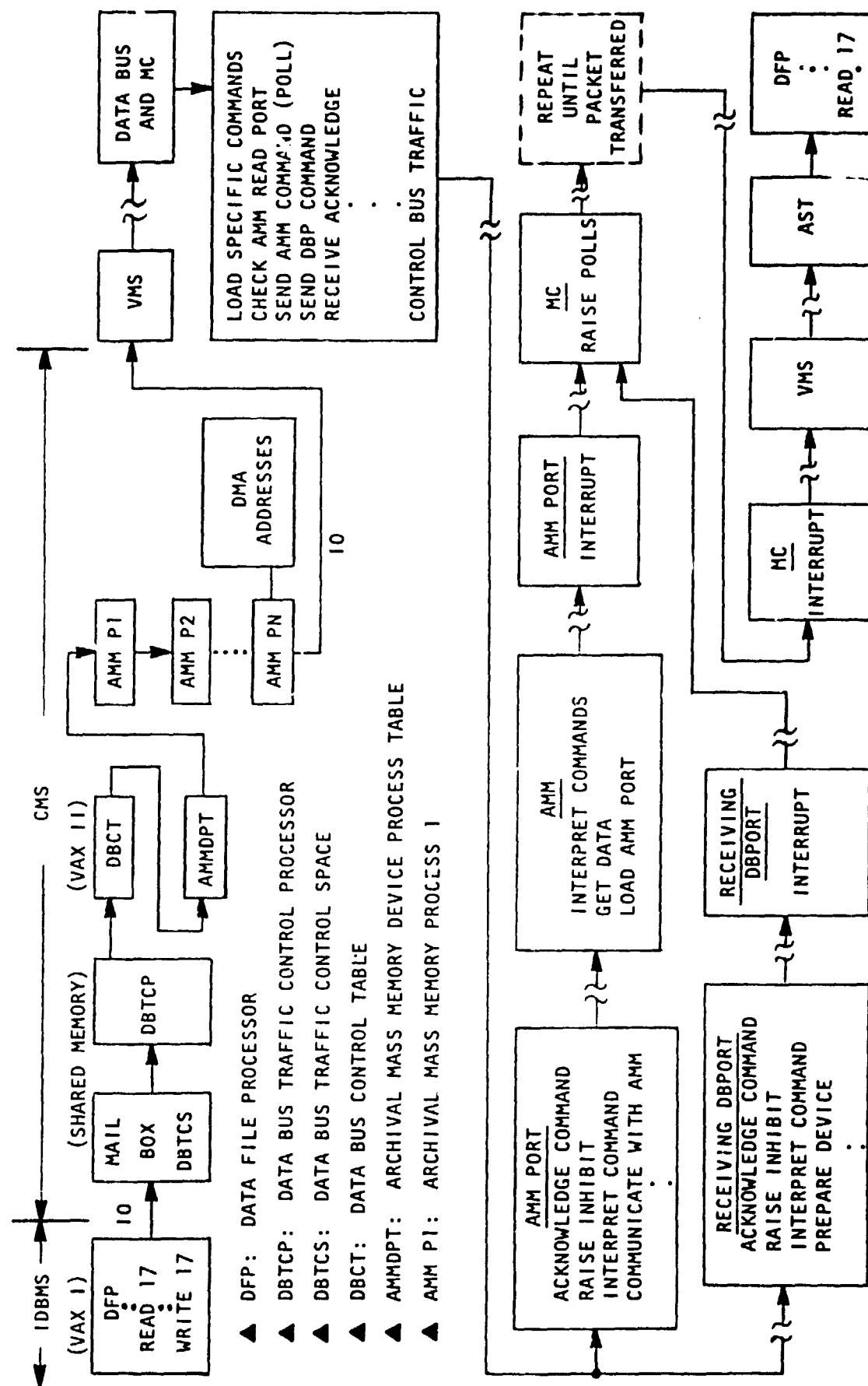


Figure 4-3. Typical Data Flow from IDBMS to Retrieve AMM Data

will set its status table to indicate a data retrieval is in process for the AMM. If the MC receives the acknowledge from the AMM port, the MC will continue to service other bus traffic while it awaits AMM retrieval. The MC will also prepare the port of the receiving device, in this case the port on VAX 11.

The AMM port will interpret the command and initiate communication with the AMM. The port will lower its inhibit as soon as its buffer is empty to permit continued communication and write commands to be received.

When the AMM retrieves the desired data, communication will be initiated with the port. The port will raise the inhibit line on its transmitting port pending the transfer of the first block of data from the AMM to the AMM port.

When the transmittal port buffer is full, the port will raise its interrupt to the MC. If the receiving port had already raised its interrupt and signified its readiness for data by dropping its inhibit (in this case, the port is integral with the MC so these are internal signals), the MC will respond during the data bus data synch time by simultaneously raising the poll lines to the AMM transmitting port and the receiving port. The block data will be transferred. The receiving port will acknowledge and the MC will acknowledge to the transmitting port. This will signal the transmitting port to raise its inhibit and fetch another block of data from the AMM. In the meantime, the MC will transfer the block of data from its buffer, over the UNIBUS, to the designated memory addresses. This transfer will be under the control of the UNIBUS and will not require the computer.

At the completion of the packet transfer, which may require several block transfers, the MC will reset its status tables. The VAX 11 VMS MC driver will perform whatever post processing that is required and the Asynchronous Services Trap (AST) will notify the calling process of the completion of the I/O process.

#### 4.8.2 TYPICAL USER TERMINAL INTERACTION WITH DBMS

A typical sequence of data flow originating from a device on a data port is illustrated in Figure 4-4. For this example, an operator at a user terminal enters a command, a user ID, and a password. The terminal device establishes the initial communication with the Data Bus Port (DBP). The DBP inhibits data flow into its buffer from the data bus while its buffer is loading from the user terminal. The Data Bus Port under the control of its local micro-processor incorporates the necessary op code and argument into its buffer. The op code signifies "establish user terminal communication." The argument identifies the user terminal or display if a multiplexer is involved, and the destination to IDBMS. The data supplied from the user terminal is "Command - User ID - Password." When the DBP buffer is filled, the DBP interrupts the master controller. The master controller responds with the sequence of steps of Table 4-2.

Table 4-2. Master Controller Interrupt Process

1. Complete current process
2. Scan interrupt register according to priority
3. Raise poll to device and MC port during control time synch
4. Receive command in 2K bus port RAM
5. Acknowledge receipt of message
6. Transfer data to microcomputer RAM
7. Interpret command
8. Set local status for device involved
9. Clear interrupt in register
10. Process command

The master controller processes the command and determines that the sequence of Table 4-3 is required. The master controller driver inputs the command to the Data Bus Traffic Control Processor (DBTCP) space. The DBTCP directs control to the Data Bus Command Processor which with the help of a Command Process Table and the User Terminal Device Process Table cause a sequence of



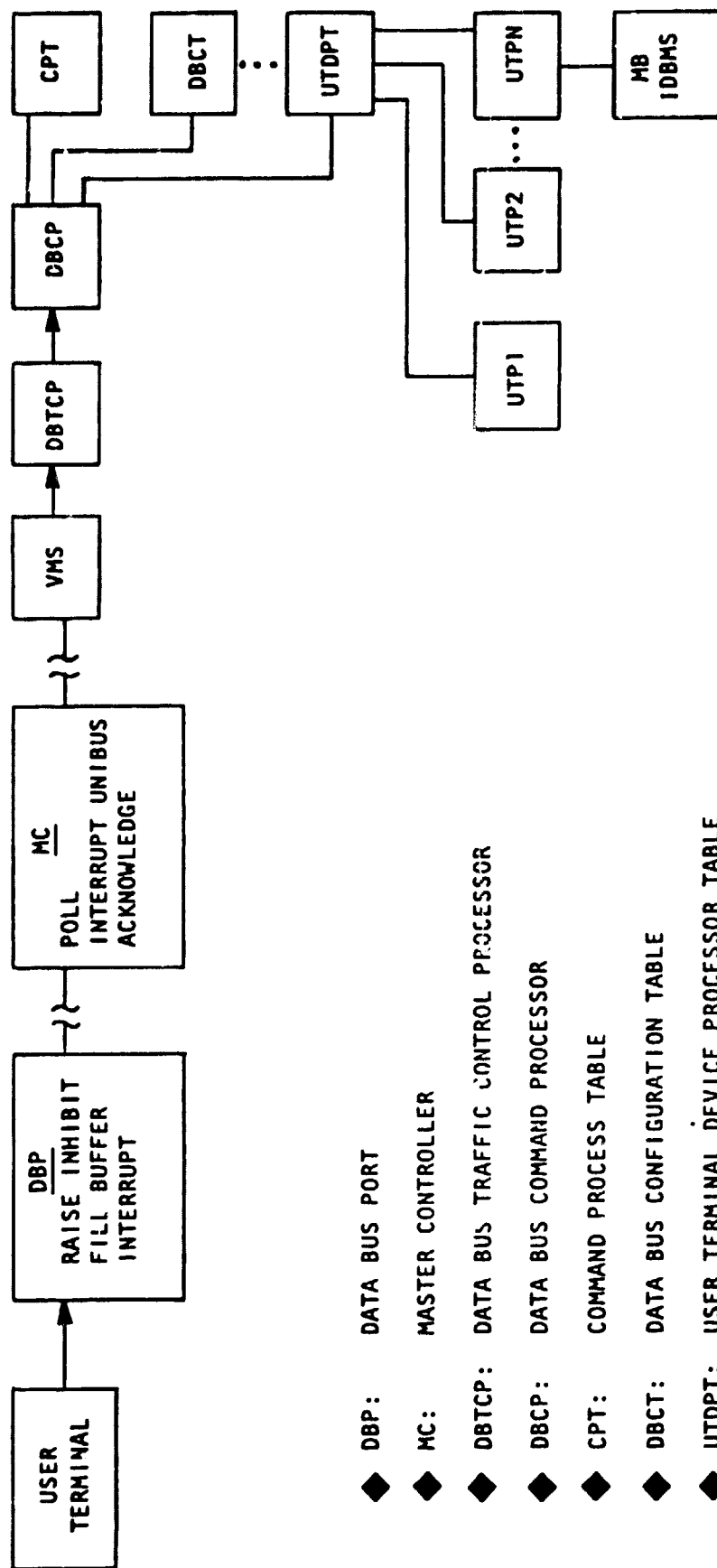


Figure 4-4. User Terminal Interaction with IDBMS

Table 4-3. Sequence for Master Controller Processing Command from UT1 to Establish Communication

1. Check status of device (VAX 1)
2. Check user terminal inhibit register
3. Raise user terminal poll during data synch time
4. Receive data in 2K RAM
5. Acknowledge receipt of data
6. Set local status
7. Establish protocol with DR11-B
8. Transfer data to DR11-B

(DR11-B and UNIBUS transfers data to user communication I/F address space just as display driver would do.)

processes to be performed. This sequence of processes is analogous to the sequence of processes normally performed by the user terminal driver if the device were physically interfaced to the VAX. Many of the processes will be identical. The completion of the process results in the data input at the user terminal being placed in the IDBMS mail box which the VMS connects to the internal read command in lieu of reading from an external I/O device.

#### 4.9 SOFTWARE SYSTEM INTERFACES

Interfaces between the IDBMS and the CMS shall be via mailboxes. As a minimum, the following four interfaces shall be implemented.

- o Notification from the CMS to the DBA that header data is in the file on auxiliary storage
- o Notification from IDBMS to CMS that a portion of the header data file may be purged.
- o Requests from IDBMS to CMS for data packets in the AMM
- o Requests from CMS to IDBMS for relational retrieval

The formats of the messages for each of the above interfaces shall be TBD.

## SECTION 5

### OTHER FACTORS

#### 5.1 RELIABILITY

Equipment reliability shall be adequate to achieve data transfers with an unrecoverable error rate of not greater than one bit error in  $10^9$  bits. Each LASER in the system shall have a minimum lifetime of  $10^4$  hours.

#### 5.2 MAINTAINABILITY

Software diagnostic shall monitor the system components and shall isolate a system fault at least to the subsystem level as defined in Section 3 of this specification. Equipment design shall permit isolation of subsystem faults to the level of a replaceable unit with standard test equipment and maintenance procedures.

#### 5.3 WORKMANSHIP

Workmanship shall conform to commonly accepted standards for commercial equipment.

#### 5.4 QUALITY ASSURANCE PROVISIONS

The contractor shall conduct a quality control program and certify that all material (hardware and software) delivered meets the requirements of this specification.

- a. Contractor inspection shall be in accordance with the provision of the contract schedule, general provisions, statement of work, this specification and applicable drawing requirements.
- b. The Government Procurement Quality Assurance (PQA) for inspection and acceptance of all equipment delivered under this specification shall be conducted in accordance with the demonstration provisions of paragraph 4.1.4 and 5.6.
- c. Final inspection and acceptance of the software shall be evidenced by the contracting officer's approval of the contractor's statement of completion and certification that it comply with all contract requirements.

- d. Inspections and acceptance of all technical data shall be accomplished in accordance with Data Items Descriptions.

## 5.5 DOCUMENTATION

### 5.5.1 SPECIALLY DEVELOPED EQUIPMENT

Specially developed equipment shall be supplied with a set of schematics and maintenance manuals. Test methods and procedures shall be included and have the capability of isolating faults to component level. Functional specifications will be included for all components. User's manuals shall be included on all equipment with user interfaces. Operation procedures shall be included with all equipment.

### 5.5.2 COMMERCIALY PURCHASED EQUIPMENT

Commercially purchased equipment shall be supplied with a set of maintenance manuals and test procedures. Test procedures shall be adequate to isolate faults down to the board level. A functional specification shall be supplied for each piece of equipment. Operation procedures shall be supplied with all equipment. User manuals shall be supplied for all equipment with user interfaces.

### 5.5.3 SOFTWARE

Software shall be supplied with readable code. All software, including purchased software shall have flow charts with detailed annotation and source code listing.

## 5.6 TESTING

Tests shall be run to insure that each module satisfies the requirements of this specification. Tests of each module shall be run at the contractor's facility with government witnesses to certify each test. Test procedures shall be included with all modules when delivered. In addition, tests will be run at the government facility on the entire DBMS to insure compatibility with this specification. A copy of the test procedure shall be included with the system.

## LIST OF ACRONYMS

AMM	Archival Mass Memory
AS	Auxiliary Storage
CCITT	Consultative Committee for International Telephony and Telegraphy
CMS	Configuration Management System
CRC	Cyclical Redundancy Check
CRT	Cathode Ray Tube
DB	Data Bus
DBMS	Data Base Management System
DBP	Data Base Processor
DEC	Digital Equipment Company
DFP	Data File Processor
DMP	Data Manager Processor
FSC	Frame Sequence Check
GFE	Government Furnished Equipment
HDM	Header Data Manager
IAS	Information Adaptive System
ID	Identification
IDBMS	Integrated Data Base Management System
IFP	Image Format Processor
IP	Instrument Packet
MC	Master Controller
MDTS	Modular Data Transport System
MID	Mission Identification
MPP	Massively Parallel Processor
NEEDS	NASA End-to-End Data System
PHI	Packet Header Interface
PL	Packet Length
PP	Packet Parity
SAI	Staging Area Interface
SHL	Secondary Header Length
SID	Source Identification
SIDP	Source ID Parity
SP	Spare
SSC	Source Sequence Count
TTL	Transistor-Transmitter Logic
UAP	User Assistance Processor
UT	User Terminal
VAX	(Not an acronym - a DEC computer model designation)